

SKRIPSI

PERANCANGAN DAN PEMBUATAN PHONE BOOK DIGITAL DAN DIALER DENGAN METODE DDS BERBASIS MIKRO AT89S8252



**Disusun Oleh :
CHOIRIL SETIAWAN
03.17.030**

**JURUSAN TEKNIK ELEKTRO S -1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

MARET 2009

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN PHONE BOOK DIGITAL DAN DIALER DENGAN METODE DDS BERBASIS MIKRO AT89S8252

SKRIPSI

*Disusun Dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

Disusun Oleh :

**CHOIRIL SETIAWAN
03.17.030**

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

**Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274**

**Ibrahim Ashari, ST, MT
NIP. Y. 1030100358**

Mengetahui

Ketua Jurusan Teknik Elektro S-1



**Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274**

**JURUSAN TEKNIK ELEKTRO S -1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

MARET 2009



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Choiril Setiawan
NIM : 03.17.030
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Masa Bimbingan : 15 Desember 2007 s/d 15 Juni 2008
Judul Skripsi : Perancangan dan Pembuatan Phone Book Digital Dan
Dialer Dengan Metode DDS Berbasis Mikro AT89S8252

Dipertahankan di hadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 24 Maret 2009
Dengan Nilai : 81,14 (A) *Ref*



(Ir. Sidik Noertjahjono, MT)
NIP.Y. 1028700163

Sekretaris Majelis Pengguji

(Ir. F. Yudi Limpraptono, MT)
NIP.Y. 1039500274

Penguji I

(Joseph Dedy Irawan, ST, MT)
NIP.Y. 132315178

Penguji II

(Bambang Prio Hartono, ST, MT)
NIP. Y. 1038400082

ABSTRAKSI

PERANCANGAN DAN PEMBUATAN PHONE BOOK DIGITAL DAN DIALER PENGGUNAKAN METODE DDS BERBASIS MIKRO AT89S8252

**Choiril Setiawan, 03.17.030, Ir. F. Yudi Limpraptono, MT
Ibrahim Ashari, ST, MT**

**Jurusan Teknik Elektro S1 – Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang**

Kata Kunci : Metode DDS, AT89S8252, AT89S51

Teknologi yang ingin dikembangkan adalah phone book digital dan dikembangkan menjadi phone book digital dan dialer menggunakan metode DDS . Rumusan masalah pada alat ini adalah bagaimana cara merancang dan membuat rangkaian phone book digital dan dialer menggunakan metode DDS dan bagaimana merancang dan membuat perangkat lunak atau software pada mikrokontroler yang mengendalikan semua kerja system,tujuan adalah merancang dan membuat phone book digital dan dialer menggunakan metode DDS.

Pada perancang ini digunakan hardware yang meliputi keypad, LCD, IC Mikrokontroler AT89S8252, dan AT89S51, LCD 26 x 2 karakter, Buffer dan Spiker. Dan software yang terdiri dari program pada mikrokontroler AT89S8252 dan AT89S51 sebagai pemroses data.

Pengembangan dari phone book digital sebelumnya adalah ditambahkan dialer dan menggunakan metode DDS karena sebelumnya menggunakan ADC atau sinyal analog diubah menjadi data digital dan DDS adalah sinyal digital diubah menjadi data analog keunggulan dari DDS sendiri mempunyai setting time/kecepatan yang cepat dan memiliki resolusi frekuensi yang halus terhadap frekuensi keluaran. Prinsip kerja dari system ini adalah tombol-tombol keypad dikan menghasilkan data input ke mikrokontroler,data input dari keypad disimpan pada mikrokontroler AT89S8252 kemudian ditampilkan pada LCD dan rangkaian ini menggunakan dua mikro untuk membedakan sinyal low dan sinyal higt kemudian setelah diproses pada mikro tegangan akan disangga dengan dua buffer digital kemudian data akan diubah menjadi data analog dengan menggunakan R2R/DAC dan sebelum menuju pada OP-AMP disangga dengan buffer analog,setelah menjadi data analog kemudian tegangan dikuatkan OP-AMP dan terakhir spaiker sebagai keluaran suara yang digunakan pada pesawat telephone rumah.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran-Mu Ya Allah yang telah memberikan rahmat dan hidayah-Nya, sehingga saya dapat menyelesaikan skripsi yang berjudul **“Perancangan Dan Pembuatan Phone Book Digital Dan Dialer Dengan Metode DDS Berbasis Mikro AT89S8252”** ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penulis menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Sidik Noertjahjono, MT selaku selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Ir. F. Yudi Limpraptono, MT selaku Dosen Pembimbing I.
5. Bapak Ibrahim Ashari, ST, MT selaku Dosen Pembimbing II.
6. Kedua orang tuaku dan kakak-kakakku yang telah memberikan do’a restu, dorongan, semangat, dan biaya.
7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu Penulis mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Maret 2009

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
BERITA ACARA UJIAN SKRIPSI	iii
ABSTRAKSI	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan	3
1.5. Metodologi Penulisan	3
1.6. Sistematika Pembahasan	4
BAB II TEORI DASAR	5
2.1. Latar Belakang DDS	5
2.1.1. Kelebihan dan Fleksibilitas DDS	5
2.1.2. R2R Digital To Analog Corverter	6
2.1.3. Blok R-2R Ledder	6
2.1.4. Bagian Penguat Tegangan	7

2.2. Sistem Mikrokontroller AT89S8252	8
2.2.1. Perangkat Keras Mikrokontroler AT89S8252	9
2.2.2. Konfigurasi Pena-Pena Mikrokontroler AT89S8252	11
2.2.3. Organisasi Memory	14
2.2.4. SFR (<i>Special Function Register</i>)	15
2.2.5. Sistem Interupsi	17
2.3. DTMF (<i>Dual Tone Multi Frekuensi</i>)	18
2.4. Keypad	19
2.5. LCD M1632	19
2.6. Mikrokontroler AT89S51	20
2.6.1. Pendahuluan	20
2.6.2. Perangkat Keras mikrokontroler AT89S51	21
2.6.3. Konfigurasi Pin-Pin Mikrokontroler AT89S51	22
2.6.4. Organisasi Memory	25
2.6.5. SFR (<i>Special Function Register</i>)	26
2.6.6. Sistem Interupsi	29
BAB III PERANCANGAN DAN PEMBUATAN ALAT	31
3.1. Pendahuluan	31
3.2. Perencanaan Perangkat Keras	33
3.2.1. Sistem Mikrokontroler AT89S8252/AT89S51	34
3.2.1.1. Konfigurasi Pin Mikrokontroler AT89S8252	34
3.2.1.2. Konfigurasi Kaki Mikrokontroler AT89S51	36
3.2.2. Rangkaian <i>Clock</i>	37
3.2.3. Rangkaian Reset	38

3.2.4. Rangkaian Antarmuka Keypad.....	40
3.2.5. Rangkaian Antarmuka Modul LCD.....	41
3.2.6. Perancangan DAC (<i>digital Analog Converter</i>)	42
3.3. Perancangan Perangkat Lunak (<i>software</i>)	44
3.3.1. Software Di Mikrokontroler (MCU).....	44
3.3.2. Diagram Alir Program Komputer	45
BAB IV PENGUJIAN ALAT	46
4.1. Pendahuluan.....	46
4.2. Pengujian Rangkaian DAC R2R	47
4.2.1. Tujuan	47
4.2.2. Peralatan Yang Digunakan.....	47
4.2.3. Langkah-Langkah Pengujian	47
4.2.4. Analisa.....	49
4.3. Pengujian Rangkaian Op Amp	52
4.3.1. Tujuan.....	52
4.3.2. Peralatan yang digunakan.....	52
4.3.3. Langkah-langkah Pengujian.....	52
4.3.4. Analisa.....	53
4.4. Pengujian Hardware	54
4.4.1. Tujuan	54
4.4.1.1 Alat-alat Yang Digunakan.....	54
4.4.1.2 Prosedur Pengujian.....	55
4.4.1.3 Hasil Pengujian	55
4.4.2. Pengujian Keypad 4x4.....	57

4.4.2.1. Tujuan	57
4.4.2.2. Alat Yang Diginakan	57
4.4.2.3. Prosedur Pengujian	58
4.4.2.4. Hasil pengujian	59
4.4.3. Pengujian buffer.....	60
4.4.3.1. Peralatan yang digunakan.....	60
4.4.3.2. Langkah pengujian buffer.....	60
4.4.3.3. Analisa.....	61
4.5. Pengujian sistem secara menyeluruh.....	62
 BAB V PENUTUP	64
5.1. Kesimpulan	64
5.2. Saran	65
 DAFTAR PUSTAKA	
 LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

2.1 Diagram Blok R-2R Ladder.....	7
2.2 Bagian Penguat Tegangan.....	7
2.3 Supply DAC.....	8
2.4 OP-AMP DAC.....	8
2.5 Diagram Blok Mikrokontroler AT89S8252	10
2.6 Konfigurasi Pena-Pena AT89S8252.....	11
2.7 Osilator Eksternal AT89S8252.....	13
2.8 Keypad Matrik 4 x 4	19
2.9 Blok Diagram Mikrokontroler AT89S51.....	22
2.10 Konfigurasi Pin-Pin AT89S51.....	22
2.11 Osilator.....	25
3.1 Diagram Blok Sistem.....	32
3.2 Rangkaian Mikrokontroler AT89S8252.....	35
3.3 Konfigurasi Kaki-Kaki AT89S51.....	36
3.4 Rangkaian <i>Clock</i>	37
3.5 Rangkaian Riset.....	39
3.6 Rangkaian Input Keypad.....	40
3.7 Rangkaian Antarmuka Modul LCD.....	41
3.8 Rangkaian DAC (<i>Digital Analog Converter</i>).....	42
3.9 Flowchart Sistem	45

4.1 Rangkaian Pengujian DAC R2R.....	48
4.2 Sinyal Output Sebelum Di Filter	51
4.3 Sinyal Output Setelah Di Filter	51
4.4 Rangkaian Pengujian Op Amp	52
4.5 Rangkaian Pengujian LCD	55
4.6 Tampilan Pengujian LCD	55
4.7 Pengukuran Tegangan Awal LCD	56
4.8 Pengukuran Tegangan Setelah Melewati Dioda.....	57
4.9 Pengujian Rangkaian Keypad.....	58
4.10 Pengecekan Jalur Keypad Dengan Multimeter	60
4.11 Rangkaian Buffer.....	61
4.12 Alat Secara Keseluruhan.....	63

DAFTAR TABEL

2.1 Fungsi Khusus Pada Port 3	12
2.2 <i>Special Function Register</i>	15
2.3 Alamat Sumber interupsi.....	17
2.4 Alokasi Frekuensi DTMF	18
2.5 Fungsi Kaki-Kaki M1632.....	20
2.6 Fungsi Khusus Pada Port 3.....	24
2.7 <i>Special Function Register</i>	27
2.8 Tingkat Prioritas Interupsi.....	30
4.1 Data Hasil Pengukuran Dan Perhitungan Rangkaian DAC R2R	50
4.2 Data Hasil Pengukuran Dan Perhitungan Rangkaian Op Amp.....	53
4.3 Hasil Pengukuran Dari Rangkaian DAC R-2R.....	52
4.4 Hasil Pengujian Keypad	59
4.5. Data hasil pengukuran dan perhitungan rangkaian buffer.....	62



BAB I

PENDAHULUAN

1. Latar Belakang

Pada perancangan dan pembuatan phone book digital dan dialer dengan menggunakan metode DDS membutuhkan hardware dan software hardware meliputi LCD, Keypad, Mikro AT89S8252 dan AT89S8252 dan software meliputi program mikro sendiri. Keunggulan phone book digital dan dialer dengan metode DDS(*direct digital synthesis*) berbasis AT89S8252 dari yang dibuat sebelumnya memakai metode DDS dan ditambah dialer, keunggulannya terdapat pada metode DDS karena mempunyai setting poin yang cepat dan mempunyai resonansi yang halus terhadap frekuensi keluaran jadi untuk menuntut perkembangan dibidang elektronika yang pesat pada era globalisasi cukup mempuni memakai metode DDS.

Di dunia elektronik modulator sudah tidak asing lagi, Secara umum modulator merupakan salah satu cara modifikasi sinyal informasi yang akan disampaikan kepada suatu alat yang berbentuk sinyal digital, yaitu pulsa yang menyatakan nilai 1 dan 0 . Sinyal digital ini tidak dapat ditransmisikan begitu saja, karena *bandwidth* (lebar pita) yang dipakai oleh sinyal digital terlalu lebar. Sinyal ini harus dimodifikasi agar ia dapat ditransmisikan. Alat pemodifikasi terhadap sinyal inilah yang dinamakan modulator. (Malvino, 2009)

Hal tersebut yang mendasari untuk mengaplikasikan suatu metode *direct digital synthesis* (DDS) untuk phone book digital pada skripsi ini, seperti diketahui banyak

sekali sistem atau metoda memproduksi suatu sinyal sinus ke dalam format digital dan kemudian melakukan suatu konversi digital-ke-analog.

Tetapi, sistem yang dapat menawarkan kecepatan tinggi terhadap frekuensi keluaran, resolusi frekuensi bagus, operasi atas suatu spektrum frekuensi yang lebar dan dengan kemajuan dalam disain teknologi proses, serta sangat ringkas dan membutuhkan sedikit daya mungkin sangat sedikit. Dalam sistem ini semua yang dibutuhkan dalam system dapat terpenuhi oleh penggunaan *direct digital synthesis (DDS)* (DELTA, 2009).

2. Rumusan Masalah

Dalam perancangan dan pembuatan phone book digital dan dialer menggunakan AT89S8252 yaitu:

1. Bagaimana merancang dan membuat rangkaian phone book digital dan dialer dengan menggunakan metode DDS.
2. Bagaimana merancang dan membuat perangkat lunak atau *software* pada mikrokontroler yang mengendalikan semua kerja sistem.

3. Tujuan

Tujuan dari pembahasan ini adalah merancang dan membuat phone book digital dan dialer dengan metode DDS.

4. Batasan Masalah

Agar pembahasan dari perancangan dan pembuatan phone book digital dan dialer dengan micro AT89S8252 ini tidak terlalu meluas maka penyusun perlu membuat batasan-batasan masalah yang meliputi :

1. Pengontrol “ phone book dan dialer “menggunakan AT89S8252.
2. Tidak membahas catu daya yang di gunakan.
3. Tidak membahas *noise-noise*

5. Metodologi Penulisan

Metode yang digunakan dalam penulisan skripsi ini adalah

1. Studi Pustaka

hardware yang digunakan meliputi LCD, Keypad, Mikro AT89S8252 dan AT89S51.

2. perancangan

Dengan melakukan penelitian secara langsung mengenai objek-objek yang berhubungan langsung dengan perencanaan alat yang akan dibuat.

3. pembuatan alat

Yaitu meliputi pembuatan PCB, perakitan komponen serta penyolderan dan pembuatan perangkat lunak.

4. pengujian alat

Dengan melakukan pengujian perblok rangkaian dan kerja seluruh sistem pada alat tersebut.

5. analisa

phone book dibuat untuk menyimpan data dan untuk mendial nomer

6. pembuktian

phone book digital dan dialer dengan menggunakan metodeDDS berjalan yang saya harapkan

7. kesimpulan

kelebihan phone book digital dan dialer terdapat pada metode DDS yang mempunyai kelebihan setting poin yang cepat dan mempunyai resolusi yang halus terhadap frekuensi keluaran

6. Sistematika Pembahasan

Sistematika pembahasan dari skripsi ini terdiri dari pokok pembahasan yang saling berkaitan antara satu dengan lainnya, yaitu :

BAB I Pendahuluan

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, sistematika pembahasan dari alat yang direncanakan.

BAB II Landasan Teori

Pada bab ini dibahas tentang teori-teori yang mendukung dalam perencanaan dan pembuatan alat ini yang meliputi micro AT89S8252.

BAB III Perencanaan Dan Pembuatan Alat

Pada bab ini dibahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*).

BAB IV Pengujian Alat

Pada bab ini dibahas tentang proses serta hasil dari pengujian alat, yang didasarkan oleh pengukuran-pengukuran dan percobaan.

BAB V Penutup

Pada bab ini akan disampaikan kesimpulan dari perencanaan dan pembuatan sistem ini.

BAB II

DASAR TEORI

2.1 Latar Belakang DDS

System DDS (*Direct Digital Synthesis*) merupakan salah satu cara untuk menghasilkan sinyal sinusoidal secara langsung. Inti dari system ini adalah arsitektur akumulator dengan resolusi mencapai mili Hertz dan frekuensi sinyal yang dihasilkan dapat diatur tergantung dari sinyal frekuensi referensi dan metode perancangan. Keluaran sistem DDS yang diproses oleh Mikrokontroler berupa sinyal digital kemudian menjadi masukan untuk DAC (*D/A converter*) dan LPF (*Low Pass Filter*) untuk menghasilkan sinyal sinusoidal yang sempurna.

Semua parameter kontrol sistem DDS berada dalam bentuk besaran digital. Sistem DDS pada dasarnya terdiri atas akumulator fasa, LUT (*Look Up Table*), dan osilator sebagai pembangkit frekuensi referensi (*clock*). Sedangkan DAC (*digital to Analog Converter*) dan LPF (*Low Pass Filter*) merupakan komponen-komponen penunjang sistem DDS.

2.1.1 Kelebihan Dan Fleksibilitas DDS

Kelebihan penggunaan Sistem DDS adalah Karakteristik sistem DDS itu sendiri, dimana keutamaan dari sistem ini adalah memiliki *settling time* kecepatan yang cepat dan memiliki resolusi frekuensi yang halus terhadap frekuensi keluaran, operasi atas suatu spektrum frekuensi yang lebar dan dengan kemajuan dalam disain teknologi proses, serta

sangat ringkas dan sedikit membutuhkan pemakain daya. sehingga sangat memungkinkan sistem DDS bisa lebih dikembangkan untuk desain alat yang berkaitan dengan aplikasi-aplikasi frequency hopping serta system-sistem yang berkaitan dengan peralatan pemancar radio, TV, peralatan test, dll.

2.1.2 R2R Digital To Analog Converter

DAC-R-2R Ledder ini memiliki kelebihan-kelebihan antara lain

- tegangan out analog yang dapat diatur antara 5V sampai 34V sesuai yang diinginkan
- arus output yang lebih besar dari pada DAC yang lain

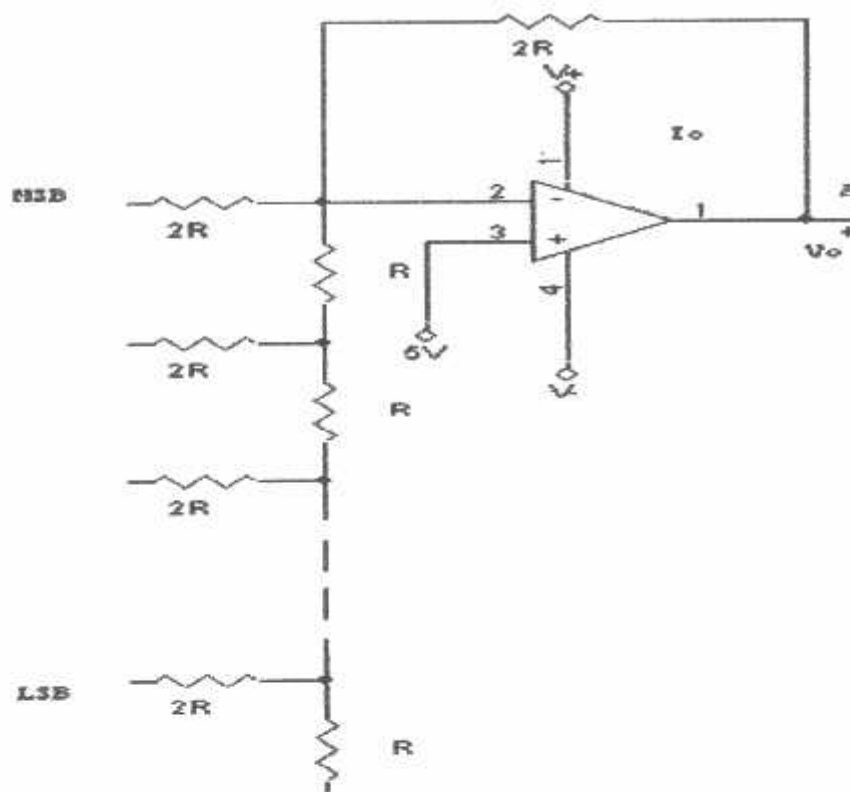
hal ini disebabkan DAC ini menggunakan OP-AMP secara umum rangkaian DAC

R-2R ledder ini dapat dibagi menjadi 2 bagian utama, yaitu :

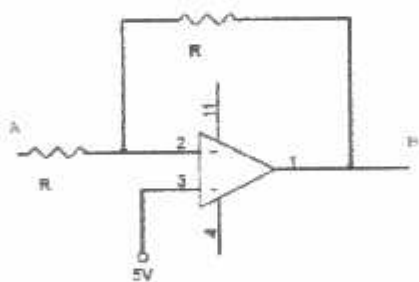
1. Blok R-2R Ledder
2. Blok penguat tegangan

Blok R-2R Ledder

Rangkain R-2R ledder mengkonversikan 8 bit data digital dari mikrokontroller menjadi tegangan analog yang bersesuaian dan apabila semua bit data berlogika 1 (FFH) maka tidak ada arus I_0 yang mengalir sehingga teg output (V_0) = 5V apabila semua bit data = 0 (0H) maka mengalir arus I_0 dan tegangan output = 10V



Pada bagian ini ada suatu level DC 5V yang harus dihilangkan agar output DAC berayun antara 0 sampai 5V, untuk itu output DAC ini diinputkan lagi ke sebuah penguat membalik yang mana akan menghasilkan tegangan output DAC 0-5V saja

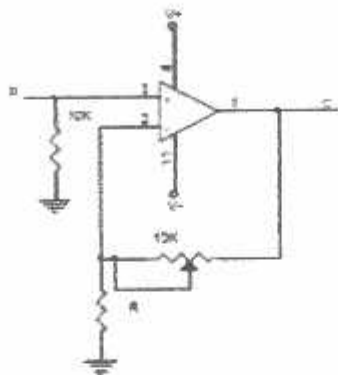


bagian penguat tegangan

Output DAC ini masih dapat kelevel yang lebih besar tergantung dari besar tegangan Supply yang digunakan. tegangan output ini maksimal 1,5V dibawah tegangan supply positifnya. Rangkaian dasar penguat tegangan adalah penguat tegangan

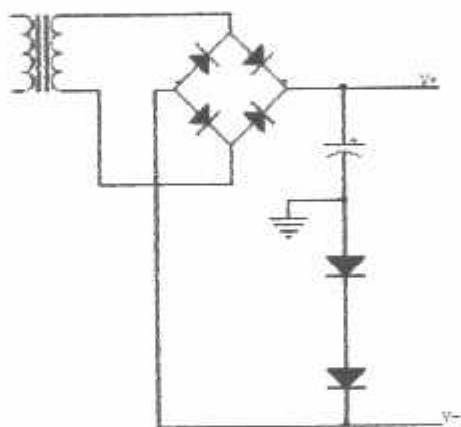
adalah penguat tegangan tidak membalik seperti pada gambar berikut

$$V_O = V_C - (1 + R_f/R) V_B$$



Pada bagian ini, tegangan $-V$ dibutuhkan untuk menghasilkan respon yang lebih linier dari OP-AMP yang digunakan.

Supply DAC



2.2. Sistem Mikrokontroler AT89S8252

Perbedaan mendasar antara mikrokontroler dan mikroprosesor adalah mikrokontroler selain memiliki CPU juga dilengkapi memori dan input output yang merupakan kelengkapan sebagai sistem minimum mikrokomputer sehingga sebuah

mikrokontroler dapat dikatakan sebagai mikrokomputer dalam keping tunggal (*Singgel Chip Mikrocomputer*) yang dapat berdiri sendiri.

Mikrokontroler AT89S8252 adalah mikrokontroler ATMEL yang kompatibel penuh dengan mikrokontroler keluarga MCS-51, membutuhkan daya rendah, memiliki performance yang tinggi dan merupakan mikrokomputer 8 bit yang dilengkapi dengan 8 KiloByte Flash memori untuk Program, 2 KiloByte EEPROM (*Electrical Eraseable and Programmable Read Only Memory*) dan 256 Byte RAM internal. Program memori yang dapat diprogram ulang dalam sistem atau menggunakan programmer nonvolatile Memori Konvensional. Dalam sistem mikrokontroler terdapat dua hal mendasar, yaitu : Perangkat lunak dan perangkat keras yang keduanya saling terkait dan mendukung.

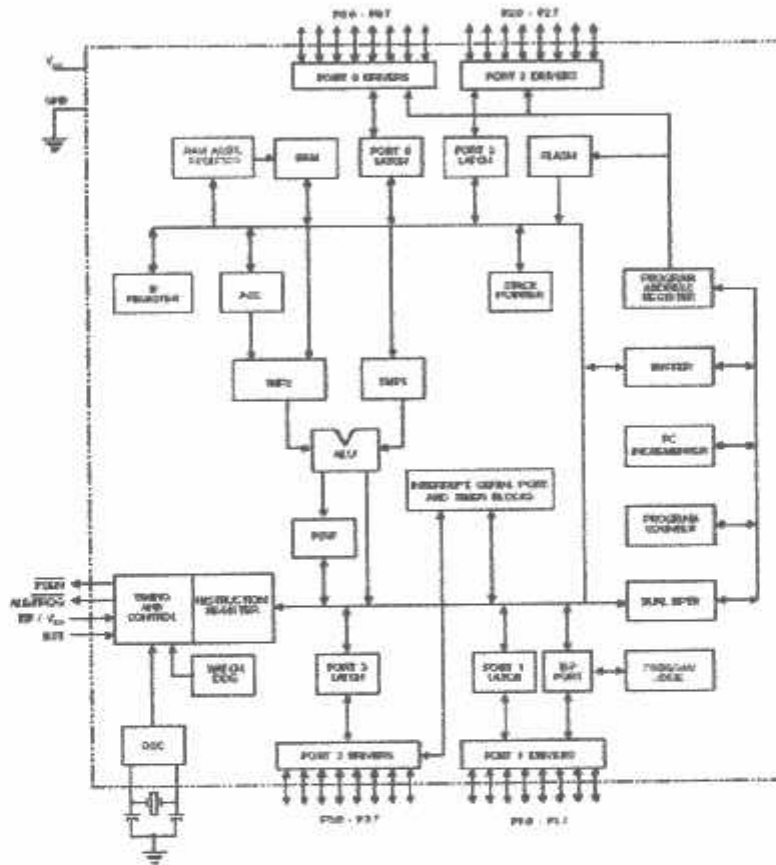
2.2.1. Perangkat Keras Mikrokontroler AT89S8252

Secara umum Mikrokontroler AT89S8252 memiliki :

- CPU 8 Bit termasuk keluarga MCS-51
- 8 Kbyte Flash Memori
- 256 Byte Internal Memori
- 32 Port I/O, Masing-masing terdiri atas 8 Jalur I/O
- 2 Timer/Counter 16 Bit
- 2 Serial Port Full Duplex
- Kecepatan pelaksanaan instruksi per siklus 1 μ S pada frekuensi clock 12 Mhz.
- 2 DPTR (*Data Pointer*)
- Watchdog Timer

- Fleksibel ISP Programming

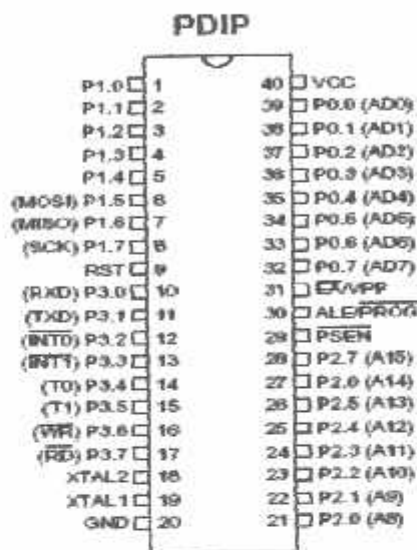
Dengan keistimewaan diatas pembuatan alat menggunakan AT89S8252 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak. Adapun blok diagram dari mikrokontroler AT89S8252 adalah sebagai berikut :



Gambar 2-1 Diagram Blok Mikrokontroler AT89S8252
(Sumber: Data Sheet Atmel AT89S8252)

2.2.2. Konfigurasi Pena-Pena Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 terdiri dari 40 pin dengan konfigurasi sebagai berikut :



Gambar 2-2 Konfigurasi Pena-pena AT89S8252
(Sumber: Data Sheet Atmel AT89S8252)

Fungsi-fungsi tiap pinnya adalah sebagai berikut :

- VCC (Supply Tegangan) , pin 40
- GND (Ground), pin 20
- Port 0, pin 32-39

Merupakan port input-output dua arah, tanpa internal pull-up dan konfigurasikan sebagai multipleks bus alamat rendah (A_0-A_7) dan data selain pengaksesan program memory dan data memory eksternal

- Port 1, pin 1-8

Merupakan port input-output dua arah dengan internal pull-up.

- Port 2, pin 21-28

Merupakan port input-output dengan internal pull-up. Mengeluarkan alamat tinggi selama pengambilan program memori eksternal.

- Port 3, pin 10-17

Merupakan port input-output dengan internal pull-up, dimana Port 3 juga memiliki fungsi khusus dan dapat dilihat pada tabel berikut

Tabel 2-1 Fungsi Khusus Pada Port 3

Nama Penyemat	Fungsi Khusus
Port 3.0	RxD (Port Masukan Serial)
Port 3.1	TxD (Port Keluaran Serial)
Port 3.2	/INT0 (Masukan Interupsi eksternal 0)
Port 3.3	/INT1 (Masukan Interupsi Eksternal 1)
Port 3.4	T0 (Masukkan Pewaktu Eksternal 0)
Port 3.5	T1 (Masukkan Pewaktu Eksternal 1)
Port 3.6	/WR (Sinyal Tulis Memori Data Eksternal)
Port 3.7	/RD (Sinyal Baca Memori Data Eksternal)

(Sumber: Data Sheet Atmel AT89S8252)

- RST (Reset), pin 9

Input Reset merupakan reset master untuk AT89S8252.

- ALE/ Prog (Address Latch Enable), pin 30

Digunakan untuk menahan alamat memori eksternal selama pelaksanaan interuksi.

- PSEN (Program Store Enable), pin 29

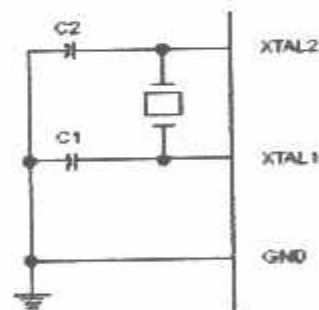
Merupakan sinyal pengontrol yang memperbolehkan program memori eksternal masuk ke dalam bus.

- EA/VPP (*External Access*), pin 31

Dapat diberikan logika rendah (*Ground*) atau logika tinggi (*5 Volt*). Jika diberikan logika tinggi maka mikrokontroler akan mengakses program dari ROM internal (*EEPROM/Flash Memori*), dan jika diberikan logika rendah maka mikrokontroler akan mengakses program dari memori eksternal.

- X-TAL 1 dan X-TAL 2, pin 19,18

Pin ini dihubungkan dengan kristal bila menggunakan osilator internal. X-TAL 1 merupakan masukan ke rangkaian osilator internal sedangkan X-TAL 2 keluaran dari rangkaian osilator internal. Untuk keperluan ini diperlukan kapasitor penstabil sebesar 30 pF . Dan nilai Dari X-TAL tersebut antara 4-24 Mhz . Untuk lebih jelasnya dapat dilihat gambar pemasangan X-TAL serta kapasitor yang digunakannya.



Gambar 2-3 Osilator Eksternal AT89S8252
(Sumber: Data Sheet Atmel AT89S8252)

2.2.3. Organisasi Memory

Organisasi memori pada mikrokontroler AT89S8252 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program. Memori program digunakan untuk menyimpan instruksi-instruksi yang akan dijalankan oleh mikrokontroler, sedangkan memori data digunakan sebagai tempat yang sedang diolah mikrokontroler.

Program mikrokontroler disimpan dalam memori program berupa ROM. Mikrokontroler AT89S8252 dilengkapi dengan ROM internal, sehingga untuk menyimpan program tidak digunakan ROM Eksternal yang terpisah dari mikrokontroler. Agar tidak menggunakan memori program eksternal, penyemat/EA dihubungkan dengan Vcc (Logika 1).

Memori program mikrokontroler menggunakan alamat 16 Bit mulai dari 0000_H – 0FFF_H sehingga kapasitas penyimpanan program maksimal adalah 4 Kbyte. Sinyal /PSEN (*Program Store Enable*) tidak digunakan jika digunakan memori program internal.

Selain Program mikrokontroler AT89S8252 juga memiliki data internal sebesar 128 Byte dan mampu mengakses memori data eksternal sebesar 64 Kbyte. Semua memori data internal dapat dialamat dengan data langsung atau tidak langsung. Ciri dari pengalamatan langsung adalah *operand* adalah alamat register yang berisi alamat data yang akan diolah. Sebagian memori tersebut dapat dialamat dengan pengalamatan register, dan sebagian lagi dapat dialamat dengan memori satu bit. Untuk Membaca data digunakan sinyal /RD sedangkan untuk menulis digunakan sinyal /WR.

2.2.4. SFR (*Special Fuction Register*)

Register Fungsi Khusus (*Special Fuction Register*) terletak pada 128 Byte bagian atas memori data internal dan berisi register-register untuk pelayanan lath port, timer, program status word, control peripheral, dan sebagainya. Alamat register fungsi khusus ini ditunjukkan pada tabel 2-2

Tabel 2-2 Special Fuction Register

Simbol	Nama Register	Alamat
ACC	Accumulator	E0 _H
B	Register B	F0 _H
PSW	Program Status Word	D0 _H
SP	Stack Pointer	81 _H
DPTR	Data Pointer 2 Byte	
DPL	Bit Rendah	82 _H
DPH	Bit Tinggi	83 _H
P0	Port 0	80 _H
P1	Port 1	90 _H
P2	Port 2	A0 _H
P3	Port 3	B0 _H
IP	Interupt Periority Control	D8 _H
IE	Interupt Enable Control	A8 _H
TMOD	Timer/Counter Mode Control	89 _H
TCON	Timer/Counter Control	88 _H
TH0	Timer/Counter 0 High Control	8C _H
TL0	Timer/Counter 0 Low Control	8A _H
TH1	Timer/Counter 1 High Control	8D _H
TL1	Timer/Counter 1 Low Control	8B _H

SCON	Serial Control	98 _H
SBUF	Serial Data Buffer	99 _H
PCON	Power Control	87 _H

(Sumber: Data Sheet Atmel AT89S8252)

Beberapa macam register fungsi khusus yang sering digunakan adalah sebagai berikut :

- *Accumulator* (ACC) merupakan register untuk penambahan dan pengurangan. Perintah *Mnemonic* untuk mengakses akumulatur disederhanakan sebagai A.
- *Register B* Merupakan register khusus yang berfungsi melayani operasi perkalian dan pembagian.
- *Stack Pointer* (SP) merupakan register 8 bit yang dapat diletakkan di alamat manapun pada RAM internal.
- *2 Data Pointer* (DPTR) terdiri atas dua register, yaitu untuk byte tinggi (*Data Pointer High*,DPH) dan byte rendah (*Data Pointer Low*,DPL) yang berfungsi untuk mengunci alamat 16 Bit.
- *Port 0 sampai Port 3* merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0,1,2,3. Masing-masing register ini dapat dialamati per-byte maupun per-bit.
- *Control Register* terdiri dari register yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu register IP (*Interupt Priority*) dan Register IE (*Interupt Enable*). Untuk mengontrol pelayanan timer/counter terdapat register khusus,yaitu register TCON

(*Timer/Counter Control*) serta pelayanan port serial menggunakan register *SCON (Serial Port Control)*.

2.2.5. Sistem Interupsi

Mikrokontroler AT89S8252 mempunyai 5 Buah sumber interupsi yang dapat membangkitkan permintaan interupsi, yaitu INT0, INT1, T1, T2 dan Port Serial.

Saat terjadinya interupsi mikrokontroler secara otomatis akan menuju ke subrutin pada alamat tersebut. Setelah interupsi selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Tiap-tiap sumber interupsi dapat *enable* atau *disable* secara software.

Tingkat prioritas semua sumber interrupt dapat diprogram sendiri-sendiri dengan set atau clear bit pada (*Interrupt Priority*). Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas yang sama diterima bersamaan, akan dilakukan polling untuk menentukan mana yang akan dilayani.

Tabel 2-3 Alamat Sumber Interupsi

Sumber Interupsi	Alamat Awal
Interrupt Luar 0 (INT 0)	03 ₁₆
Pewaktu /Pencacah 0 (T0)	0B ₁₆
Interrupt Luar 1 (INT 1)	13 ₁₆
Pewaktu /Pencacah 1 (T1)	1B ₁₆
Port Serial	23 ₁₆

(Sumber: Data Sheet Atmel AT89S8252)

2.3 DTMF (Dual Tone Multi Frekuensi)

Dual tone multi frekuensi (DTMF) merupakan metode yang paling banyak digunakan oleh industri telekomunikasi. Sebagai besar industri telekomunikasi menggunakan sistem DTMF sebagai tone dialer karena mempunyai kemampuan dengan mengirim sinyal pada transmisi suara dan mempunyai kecepatan dial.

DTMF menggunakan 16 buah frekuensi suara yang merupakan sistem sinyal frekuensi ganda dalam sistem ini, sinyal (gelombang sinusoidal) tersusun atas sebuah kombinasi dari dua buah frekuensi yaitu bagian *low group* dan *high group*, yang dipilih satu demi satu pada setiap frekuensi (lebih tinggi atau lebih rendah), seperti pada table 2-4, frekuensi tersebut masing-masing dialokasikan dan hubungan dengan sebuah angka atau sebuah kode.

Tabel 2.4 alokasi frekuensi DTMF

Frekuensi rendah (Hz)	Frekuensi tinggi (Hz)			
	1209	1336	1477	1633
679	1	2	3	A
770	4	5	6	B
825	7	8	9	C
941	*	0	#	D

2.4 Keypad

Keypad merupakan salah satu sarana untuk memasukkan suatu data ke komputer atau minimum sistem. Untuk rangkaian keypad dalam tugas akhir ini digunakan keypad matrik 4 x 4.

Keypad matrik 4 x 4 merupakan susunan 16 tombol membentuk keypad sebagai sarana masukan ke mikrokontroller, meskipun jumlah tombol ada 16 tapi hanya memerlukan 7 jalur port pararel, seperti terlihat dalam Gambar 2.21.



Gambar 2.4 Keypad Matrik 4 x 4

(Sumber: Data Sheet Keypad)

2.5 LCD M1632

LCD M1632 adalah sebuah modul LCD dot matrix dengan kebutuhan daya kecil, dilengkapi dengan panel LCD bertingkat kontras cukup tinggi dan pengontrol LCD internal. Pengontrol mempunyai pembangkit karakter ROM/RAM dan penampil data RAM. Semua fungsi penampil diatur oleh instruksi-instruksi dan modul dapat dengan mudah diantar-mukakan dengan unit mikroprosesor. Keistimewaan modul LCD M1632 adalah :

- Dua baris tampilan LCD yang terdiri dari 5X7 dot matrix ditambah kursor, masing-masing baris 16 karakter.
- ROM membangkitkan 192 tipe karakter.

- RAM untuk data penampil sebanyak 80X8 bit (maksimum 80 karakter).
- Osilator internal.
- Catu daya +5 Volt.
- Secara otomatis akan reset saat catu daya dinyalakan.

LCD mempunyai 16 kaki. Fungsi tiap-tiap kaki ditunjukkan dalam Tabel 2.5

Tabel 2.5 Fungsi Kaki-Kaki M1632

No	Nama Penyemat	Fungsi
1	Vss	0 V (GND)
2	Vcc	5V \pm 10%
3	Vee	Drive LCD
4	RS	Pemilih Register 0 = Instruksi register (tulis) Bussy flag dan address counter (baca) 1 = Register data (baca dan tulis)
5	R/W	Sinyal pemilih baca dan tulis 0 = tulis 1 = baca
6	Enable	Sinyal untuk mengawali operasi
7-14	Data Bus	Saluran data
15	V + BL	Pengendali kecerahan latar belakang LCD, 4-4,42 V dan 50-200 mA
16	V - BL	Pengendali kecerahan latar belakang LCD, 0 V

(Sumber : Seiko : 1987)

2.6 Mikrokontroler AT89S51.

2.6.1. Pendahuluan

Perbedaan mendasar antara mikrokontroler dan mikroprosesor adalah mikrokontroler selain memiliki CPU juga dilengkapi memori dan input output yang merupakan kelengkapan sebagai sistem minimum mikrokomputer, sehingga sebuah mikrokontroler dapat dikatakan sebagai mikrokomputer dalam keping tunggal (*Single Chip Microcomputer*) yang dapat berdiri sendiri.

Mikrokontroler AT89S51 adalah mikrokontroler ATMEL yang kompatibel penuh dengan mikrokontroler keluarga MCS – 51, membutuhkan daya rendah, memiliki performance yang tinggi dan merupakan mikrokomputer 8 bit yang dilengkapi 4 Kbyte EEPROM (*Electrical Erasable and Programmable Read Only Memory*) dan 128 Byte RAM internal. Program memory dapat diprogram berulang – ulang atau dengan menggunakan Programmer Nonvolatile Memory.

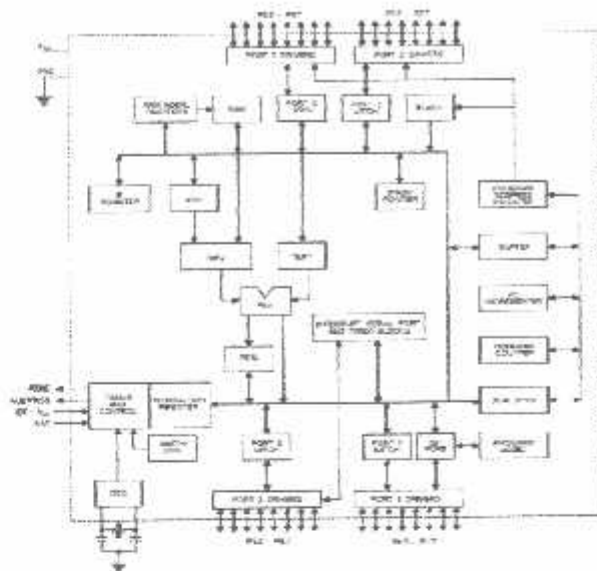
Dalam sistem mikrokontroler terdapat dua hal yang mendasar, yaitu: perangkat lunak dan perangkat keras yang keduanya saling terkait dan mendukung.

2.6.2. Perangkat keras mikrokontroler AT89S51

Secara umum Mikrokontroler AT89S51 memiliki :

- CPU 8 bit termasuk keluarga MCS-51
- 4 Kb Flash memory
- 128 byte Internal RAM
- 4 buah Port I/O, masing – masing terdiri atas 8 jalur I/O
- 2 Timer/ counter 16 bit
- 1 Serial Port Full Duplex
- Kecepatan pelaksanaan intruksi per siklus 1 μ s pada frekuensi clock 12 Mhz

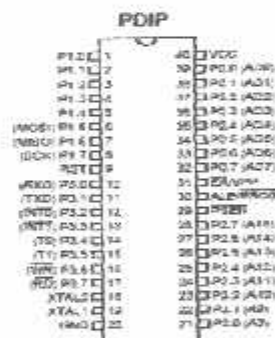
Dengan keistimewaan diatas pembuatan alat menggunakan AT89S51 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak. Adapun Blok Diagram dari Mikrokontroler AT89S51 adalah sebagai berikut:



Gambar 2.5 Blok Diagram Mikrokontroler AT89S51
(Sumber: Data Sheet Atmel AT89S8251)

2.6.3. Konfigurasi Pin-Pin Mikrokontroler AT89S51.

Mikrokontroler AT89S51 terdiri dari 40 pin dengan konfigurasi sebagai berikut:



Gambar 2.6 Konfigurasi Pin-Pin AT89S51
(Sumber: Data Sheet Atmel AT89S8252)

Fungsi tiap pin-nya adalah sebagai berikut :

1. **GND (Pin 20)**

Dihubungkan dengan Ground Rangkaian.

2. **VCC (Pin 40)**

Dihubungkan dengan sumber tegangan +5V.

3. **Port 0 (P0.0-P0.7) (Pin 32-39)**

Port 0 (P0.0 – P0.7) merupakan port I/O 8 bit dua arah. Port ini digunakan sebagai multipleks bus alamat rendah (A0 – A7) dan bus data selama pengaksesan ke memori eksternal.

4. **Port 1 (P1.0 -P1.7) (Pin 1-8)**

Merupakan port input – output dua arah dengan *pull-up*. Port ini berfungsi sebagai input atau output dan bekerja baik untuk operasi bit maupun byte, tergantung dari pengaturan software.

5. **Port 2 (P2.0 -P2.7) (Pin 21-28)**

Port 2 (P2.0 – P2.7) merupakan input – output dua arah dengan *pull-up*. Port 2 mengeluarkan *high order address byte* selama pengambilan (*fetch*) program memori eksternal dan selama mengakses data memori eksternal. Port 2 juga menerima *high order address bit* dan beberapa sinyal kontrol selama pemrograman dan verifikasi.

6. **Port 3 (P3.0-P3.7) (Pin 10-17)**

Merupakan port input-output dengan internal pull-up, dimana Port 3 juga memiliki fungsi khusus dan dapat dilihat pada tabel berikut ini:

Tabel 2.6 Fungsi Khusus Pada Port 3.

Pin Port	Fungsi Khusus
Port 3.0	RxD (Port masukan serial)
Port 3.1	TxD (Port keluaran Serial)
Port 3.2	$\overline{\text{INT0}}$ (Masukan Interupsi Eksternal 0)
Port 3.3	$\overline{\text{INT1}}$ (Masukan Interupsi Eksternal 1)
Port 3.4	I0 (Masukan Pewaktu Eksternal 0)
Port 3.5	T1 (Masukan Pewaktu Eksternal 1)
Port 3.6	$\overline{\text{WR}}$ (sinyal tulis memori data eksternal)
Port 3.7	$\overline{\text{RW}}$ (sinyal baca memori data eksternal)

(Sumber: Data Sheet Atmel AT89S8251)

6. **RST (Reset)**, pin 9.

Input reset merupakan reset master untuk AT89S51.

7. **ALE / Prog (Address Latch Enable)**, pin 30.

Digunakan untuk menahan alamat memori eksternal selama pelaksanaan intruksi.

8. **PSEN (Program Strobe Enable)**, pin 29.

Merupakan sinyal pengontrol yang memperbolehkan program memori eksternal masuk kedalam bus.

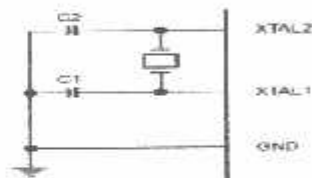
9. **EA / VPP (External Access)**, pin 31.

Dapat diberikan logika rendah (Ground) atau logika tinggi (+5V). Jika diberikan logika tinggi maka mikrokontroler akan mengakses program dari ROM internal

(EEPROM/Flash Memori), dan jika diberikan logika rendah maka mikrokontroler akan mengakses program dari memori eksternal.

10. **X-TAL 1 dan X-TAL 2**, pin 19, 18.

Pin ini dihubungkan dengan kristal bila menggunakan osilator internal. X-TAL 1 merupakan masukan ke rangkaian osilator internal sedangkan X-TAL 2 keluaran dari rangkaian osilator internal. Untuk keperluan ini diperlukan kapasitor penstabil sebesar 30pF. Dan nilai dari X-TAL. tersebut antara 4 – 24 Mhz. Untuk lebih jelasnya dapat dilihat gambar pemasangan X-TAL serta kapasitor yang digunakannya



Gambar 2.8 Osilator AT89S51
(Sumber: Data Sheet Atmel AT89S8251)

2.6.4. Organisasi Memory.

Organisasi memori pada mikrokontroler AT89S51 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program. Memori program digunakan untuk menyimpan instruksi-instruksi yang akan dijalankan oleh mikrokontroler, sedangkan memori data digunakan sebagai tempat yang sedang diolah mikrokontroler.

Program mikrokontroler disimpan dalam memori program berupa ROM. Mikrokontroler 89S51 dilengkapi dengan ROM internal, sehingga untuk menyimpan

program tidak digunakan ROM eksternal yang terpisah dari mikrokontroler. Agar tidak menggunakan memori program eksternal, EA (*Eksternal Address enable*) dihubungkan dengan Vcc.

Memori program mikrokontroler menggunakan alamat 16 bit mulai 0000H-FFFFH, sehingga kapasitas penyimpanan program maksimal adalah 4Kb. Sinyal /PSEN (*Program Store Enable*) tidak digunakan jika digunakan memori program internal.

Selain memory program mikrokontroler 89S51 juga memiliki data internal 128 *byte* dan mampu mengakses memori data eksternal sebesar 64 Kb. Semua memori data internal dapat dialamati dengan data langsung atau tidak langsung. Ciri dari pengalamatan langsung adalah *operand* alamat *register* yang berisi alamat data yang akan diolah. Sebagian memori tersebut dapat dialamati dengan pengalamatan *register*, dan sebagian lagi dapat dialamati dengan memori satu bit. Untuk membaca data digunakan sinyal /RD sedangkan untuk menulis digunakan sinyal /RW.

2.6.5. SFR (*Special Function Register*).

Register fungsi khusus (*Special Function Register*) terletak pada 128 byte bagian atas memori data internal dan berisi register-register untuk pelayanan latch port, timer, program status words, control peripheral dan sebagainya. Alamat register fungsi khusus ditunjukkan pada tabel 2.7.

Tabel 2.7 Special Function Register

Simbol	Nama Register	Alamat
ACC	Accumulator	E0 _H
B	Register B	F0 _H
PSW	Program Status Word	D0 _H
SP	Stack Pointer	81 _H
DPL	Bit rendah	82 _H
DPH	Bit Tinggi	83 _H
P0	Port 0	80 _{FF}
P1	Port 1	90 _H
P2	Port 2	A0 _H
P3	Port 3	B0 _H
IP	Interrupt Periority Control	B8 _H
IE	Interrupt Enable Control	A8 _H
TMOD	Timer/Counter Mode Control	89 _H
TCON	Timer/Counter Control	88 _H
TH0	Timer/Counter 0 High Control	8C _H
TL0	Timer/Counter 0 Low Control	8A _H
TH1	Timer/Counter 1 High Control	8D _H
TL1	Timer/Counter 1 Low Control	8B _H
SCON	Serial Control	98 _H
SBUF	Serial Data Buffer	99 _H

PCON	Power Control	87 _H
------	---------------	-----------------

(Sumber: Data Sheet Atmel AT89S8251)

Beberapa macam register fungsi khusus yang sering digunakan adalah sebagai berikut ini:

- *Accumulator (ACC)* merupakan register untuk penambahan dan pengurangan. Perintah *mnemonic* untuk mengakses akumulator disederhanakan sebagai A.
- *Register B* merupakan register khusus yang berfungsi melayani operasi perkalian dan pembagian.
- *Stack Pointer (SP)* merupakan register 8 bit yang dapat diletakkan di alamat manapun pada RAM internal.
- *Data Pointer (DPTR)* terdiri dari dua register, yaitu untuk byte tinggi (*Data Pointer High, DPH*) dan byte rendah (*Data Pointer Low, DPL*) yang berfungsi untuk mengunci alamat 16 bit.
- *Port 0* sampai *Port 3* merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0, 1, 2, 3. Masing-masing register ini dapat dialamati per-byte maupun per-bit.
- *Control Register* terdiri dari register yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu register IP (*Interrupt Priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan timer/counter terdapat register khusus, yaitu register TCON (*timer/counter control*) serta pelayanan port serial menggunakan register SCON (*Serial Port Control*).

2.6.6. Sistem Interupsi.

Mikrokontroler AT89S51 mempunyai 5 buah sumber interupsi yang dapat membangkitkan permintaan interupsi, yaitu INT0, INT1, T1, T2 dan Port Serial. Saat terjadinya interupsi mikrokontroler secara otomatis akan menuju ke subrutin pada alamat tersebut. Setelah interupsi selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Tiap-tiap sumber interupsi dapat enable atau disable secara software.

Tingkat prioritas semua sumber *interrupt* dapat diprogram sendiri-sendiri dengan *set* atau *clear* bit pada (*Interrupt Priority*). Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas tertinggi yang akan dilayani. Jika permintaan interupsi dengan prioritas yang sama diterima bersamaan, akan dilakukan polling untuk menentukan mana yang akan dilayani. Bit-bit pada IP adalah sebagai berikut:

-	-	-	PS	PT1	PX1	PT0	PX0
---	---	---	----	-----	-----	-----	-----

Priority bit = 1 menandakan prioritas tinggi

Priority bit = 0 menandakan prioritas rendah

Simbol	Posisi	Fungsi
-	IP.7	Kosong
-	IP.6	Kosong
-	IP.5	Kosong
PS	IP.4	Bit prioritas interupsi port serial
PT1	IP.3	Bit prioritas interupsi Timer 1
PX1	IP.2	Bit prioritas interupsi

PT0	IP.1	Bit prioritas interupsi Timer 0
PX0	IP.0	Bit prioritas interupsi

Tabel 2.8 Tingkatan Prioritas Interupsi.

Prioritas Interupsi	Sumber Interupsi	Alamat Vektor
1	IE0 (Interupsi eksternal 0)	0003 _H
2	TF0 (timer overflow flag 0)	000B _H
3	IE1 (interupsi eksternal 1)	0013 _H
4	TF1 (timer overflow flag 1)	001B _H
5	R1 dan T1	0023 _H

(Sumber: Data Sheet Atmel AT89S8251)

BAB III

PERENCANAAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

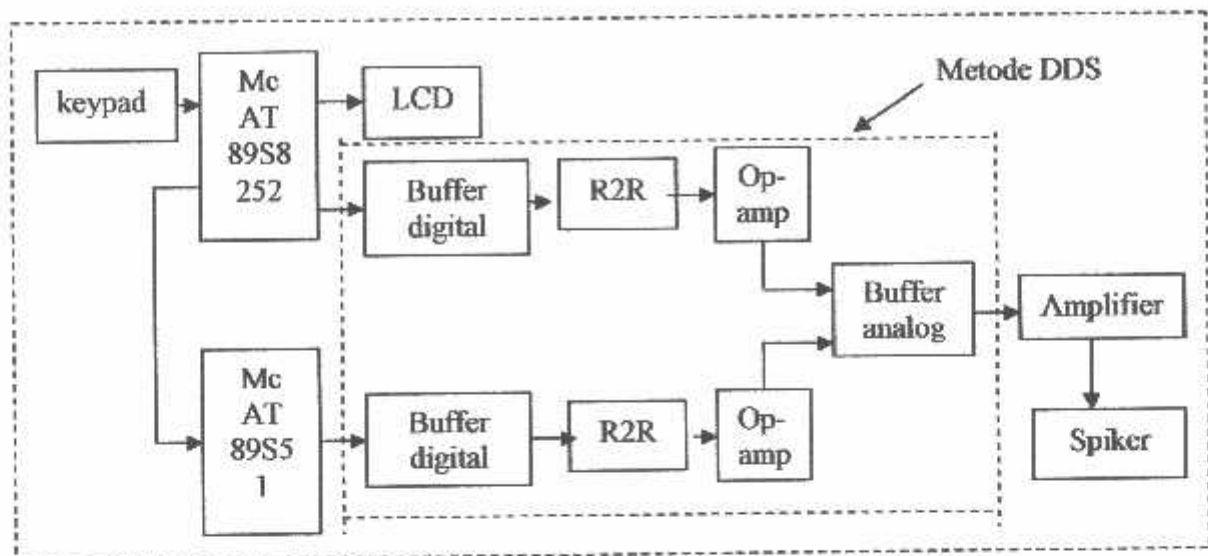
3.1. Pendahuluan

Dalam bab II telah dijelaskan bahwa *PHONE BOOK* digital dengan menggunakan Metode DDS mempunyai spesifikasi sinyal yang telah ditentukan. Oleh sebab itu setiap peralatan tambahan pada sistem DDS harus mengikuti spesifikasi yang telah ditentukan. Bab ini membahas tentang aplikasi *mikrokontroler* AT89S8252 dan AT89S51 sebagai alat memproses data-data masukan dan mengolahnya untuk dijadikan proses pada keluaran sesuai dengan perencanaan program (Sebagai pengolah data). Aplikasi tersebut meliputi spesifikasi alat, dan blok-blok rangkaian penyusun sistem.

Spesifikasi rangkaian secara global ditetapkan terlebih dahulu untuk acuan dalam perencanaan selanjutnya. Spesifikasi yang direncanakan adalah sebagai berikut:

1. Digunakan sebagai *PHONE BOOK* digital dengan menggunakan metode DDS
2. Menggunakan mikrokontroler AT89S8252/AT89S51 sebagai pengolah utama

Sistem alat terdiri atas mikrokontroler AT89S8252, AT89S51 dan komponen pendukung seperti Komunikasi Serial, DAC (Digital to Analog Converter), Diagram blok rangkaian *phone book* digital dengan Metode *Direct Digital Syntheser* (DDS) yang direncanakan ditunjukkan dalam **Gambar 3.1**



Gambar 3.1 Diagram Blok Sistem

Fungsi dari masing – masing blok diagram adalah sebagai berikut :

1. *Keypad*

Keypad yang digunakan berupa *keypad* matrik 4 x 4 yang berfungsi sebagai inputan nama dan nomor yang akan dimasukkan pada *Phone book*.

2. *LCD*

Berfungsi untuk menampilkan input dari *keypad*.

3. *Mikrokontroler AT89S51*

Memproses data dari *out put* AT89S8252 yang akan diubah menjadi data *analog*

4. *Mikrokontroler AT89S8252*

Memproses data dan menyimpan data dari input *keypad* yang akan diubah menjadi data *analog*

5. *Buffer*

Penguat / penyangga tegangan

6. *Amplifier*

Penguat tegangan yang akan dikeluarkan pada spiker

7. *Speaker*

Untuk mengeluarkan nada tone

• **Prinsip Kerja Sistem**

Adapun prinsip kerja dari sistem ini adalah sebagai berikut :

Tombol-tombol *keypad* ditekan, menghasilkan data *input* ke mikrokontroler. Data *input* dari *keypad*, di simpan pada AT89S8252 dan ditampilkan pada LCD. Setelah itu, menggunakan dua mikro AT89S8252 dan AT89S51, fungsi AT89S8252 selain memproses data juga berfungsi menyimpan data, kenapa menggunakan dua *mikro* untuk membedakan sinyal *low* dan *high* kemudian setelah diproses pada *mikro* tegangan diperkuat oleh *buffer* digital yang kemudian oleh R2R/DAC diubah dari data digital menjadi data analog dan sinyal dikuatkan lagi dengan *buffer* analog setelah data menjadi analog kemudian data tersebut diubah dalam bentuk dual *tone* yang dikeluarkan oleh spiker yang terlebih dahulu dikuatkan oleh *amplifier*.

3. 2 Perencanaan Perangkat Keras

Bagian ini menguraikan perencanaan perangkat keras yang meliputi perencanaan :

1. Sistem Mikrokontroler AT89S8252
2. sistem metode DDS
3. Antarmuka Mikrokontroler ke *keypad*
4. Antarmuka Mikrokontroler ke modul *LCD*

3.2.1 Sistem Mikrokontroler AT89S8252/AT89S51

3.2.1.1 Konfigurasi Pin Mikrokontroler AT89S8252

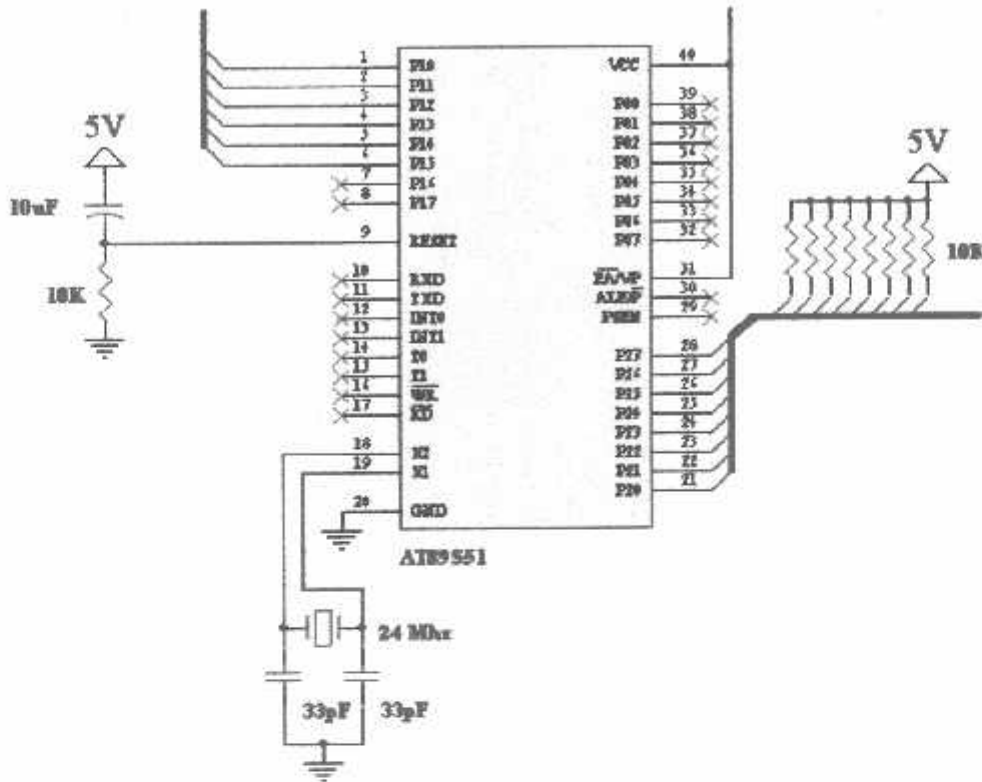
Penentuan port-port dan sinyal-sinyal yang digunakan untuk mendukung proses yang akan dilakukan adalah hal yang sangat penting. Perancangan alat ini menggunakan Mikrokontroler AT89S8252 yang merupakan suatu chip *IC* yang terdiri dari 40 pin, dalam perancangan alat ini pin-pin yang digunakan adalah sebagai berikut :

1. Pin 1-8 / digunakan sebagai keluaran (*output*) modul keypad
2. Pin 9 / Port (RST) digunakan sebagai *reset system*.
3. Pin 10-11 / Port (RXD/TXD) digunakan LCD pada pin (4 dan 6).
4. Pin 12-17 / Port menuju pada pin1-6 pada mikro AT89S51
5. Pin 18-19 / Port(XTAL1-XTAL2) digunakan sebagai sistem *clock*.
6. Pin 20 / port menuju *ground*
7. Pin 20-28 / port (2.1-2.7) digunakan sebagai *output buffer* analog
8. Pin 32-39 / port (0.0-0.7) digunakan sebagai *output* LCD

Adapun gambar rangkaian mikro AT89S8252 dapat dilihat pada **Gambar 3-2** :



3.2. 1.2 Konfigurasi Kaki Mikrokontroler AT89S51



Gambar 3.3 Konfigurasi Kaki-Kaki AT89S51

Dari 40 pin di atas, yang kita pergunakan adalah :

1. Pin 1-6 / port (P0-P5) digunakan sebagai *inputan* AT89S8252
2. Pin 9 / Port (RST) digunakan sebagai *reset system*
3. Pin 18-19 / Port(XTAL1-XTAL2) digunakan sebagai sistem *clock*
4. Pin 20-28 / port (P2-P7) digunakan sebagai *output buffer analog*

Gambar 3-3 merupakan konfigurasi kaki-kaki AT89S51

3.2.2 Rangkaian Clock

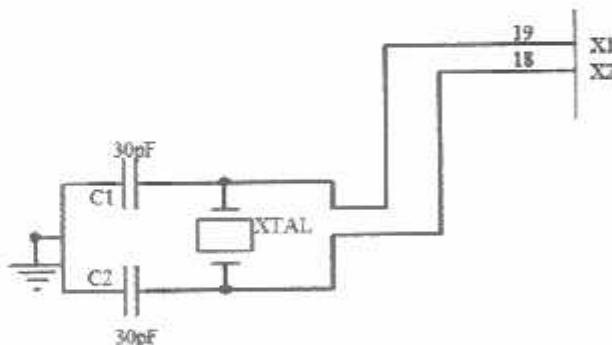
Kecepatan proses pengolahan data pada mikrokontroler ditentukan oleh *Clock* (pewaktu) yang dikendalikan oleh mikrokontroler tersebut. Pada Mikrokontroler AT89S51 terdapat *internal clock* generator yang berfungsi sebagai sumber *clock*, tapi masih memerlukan rangkaian tambahan untuk membangkitkan *clock* yang diperlukan.

Rangkaian *clock* ini terdiri atas dua buah kapasitor, dan sebuah Kristal yang terangkai sedemikian rupa dan kemudian dihubungkan dengan chip yang tersedia pada AT89S51.

Dalam perencanaan ini menggunakan :

- ✓ $C = 30 \text{ pF}$. Penentuan besarnya kapasitansi disesuaikan dengan spesifikasi lembar data AT89S51
- ✓ X-Tal 12 MHz.

Adapun gambar rangkaian *clock* dapat dilihat pada gambar 3-4:



Gambar 3.4 Rangkaian Clock

Dengan menggunakan nilai kristal diatas maka dapat dihitung waktu yang diperlukan untuk satu siklus mesin.

Diketahui : $F = 12 \text{ Mhz}$

$$T = \frac{1}{f}$$

Sehingga :

$$T = \frac{1}{12 \text{ MHz}} = \frac{1}{12} \mu s$$

Maka untuk satu siklus mesin dari mikrokontroller AT89S51 adalah sebesar :

$$T_{me} = 12 \times T$$

$$T_{me} = 12 \times \frac{1}{12} \mu s = 1 \mu s$$

$$T_{me} = 1 \mu s$$

3.2.3 Rangkaian Reset

Rangkaian reset dalam mikrokontroller AT89S51 akan melakukan reset setelah catu daya dihidupkan. Pada saat kondisi reset maka faktor reset pada alamat 0000H akan dituju oleh mikrokontroller AT89S51 (dalam hal ini program counter) agar program yang terdapat didalam mikrokontroller kembali ke kondisi semula atau dengan kata lain mikrokontroller mengakses awal dari program yang telah diisi didalamnya. Didalam reset ini akan menggunakan beberapa macam cara untuk mereset mikrokontroller AT89S51. Cara pertama menggunakan Switch (manual), dimana user yang akan mengoperasikan switch ini. Cara ke dua menggunakan capasitor 47 uF, dimana capasitor tersebut akan berkondisi aktif high selama beberapa detik.

Besarnya nilai tahanan dan kapasitor pada rangkaian reset akan menentukan lamanya waktu pulsa reset.

Dengan rumus :

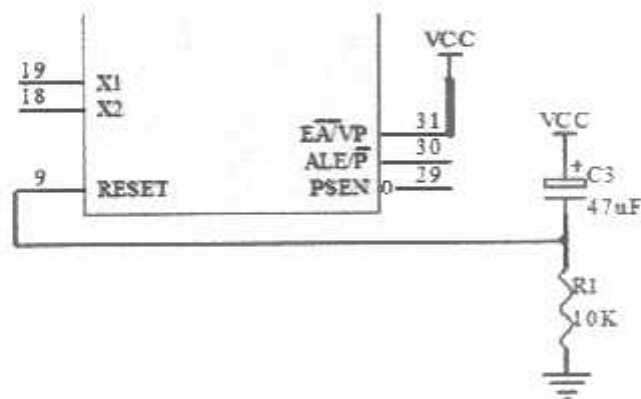
$$t = R \times C$$

Agar reset dapat terjadi secara normal maka nilai 't' harus jauh lebih besar dari waktu satu kali siklus mesin. Dengan mengambil nilai R dan C sebesar $10K\Omega$ dan $47\mu F$ maka besarnya 't' dapat dicari sebagai berikut :

$$\begin{aligned} t &= R \times C \\ &= 10K\Omega \times 47\mu F \\ &= (10 \times 10^3) \times (47 \times 10^{-6}) = 470 \times 10^{-3} \\ t &= 0,47 \text{ second} \end{aligned}$$

Maka dengan demikian ;

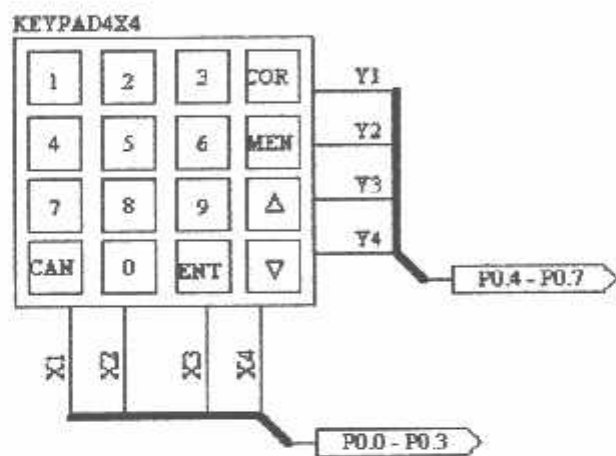
Nilai 't' jauh lebih besar dibandingkan dengan nilai minimalnya seperti pada gambar 3-5 :



Gambar 3.5 Rangkaian Reset

3.3.4 Rangkaian Antarmuka Keypad

Agar petugas (*administrator*) dapat memasukkan password dan informasi-informasi lainnya, maka diperlukan rangkaian input kontrol yang berupa rangkaian input keypad. Rangkaian input keypad ini terdapat dalam **gambar 3-6**



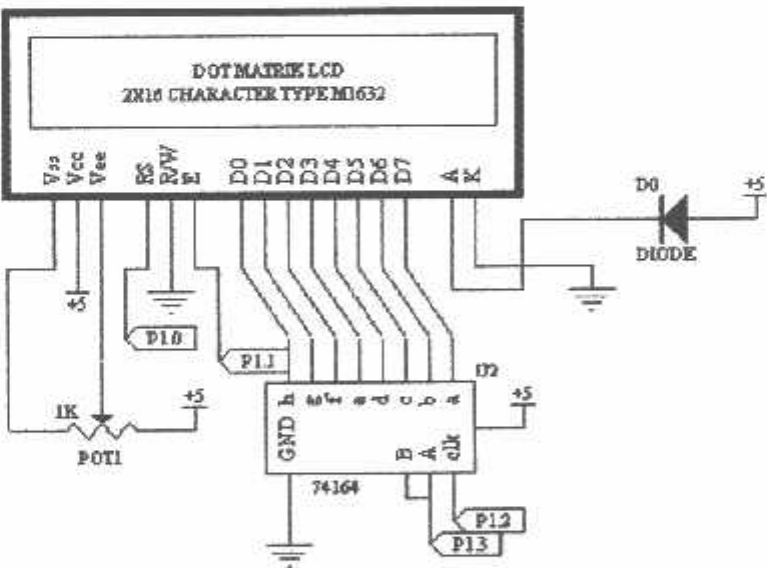
Gambar 3-6 Rangkaian Input Keypad

Komponen pendukung dalam input keypad terdiri dari 16 switch push button. Keypad ini memiliki 4 buah baris dan 4 buah kolom. Apabila terjadi penekanan tombol maka data yang dihasilkan dalam bentuk hexadesimal akan diterjemahkan oleh mikrokontroler menjadi desimal.

3.3.5 Rangkaian Antarmuka Modul LCD (*Liquid Crystal Display*)

Dalam aplikasi ini menggunakan sebuah layar LCD (*Liquid Crystal Display*) yaitu jenis M1632 yang merupakan LCD dua baris dengan setiap barisnya terdiri atas 16 karakter. Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih ter-*multiplex* dengan bus alamat. Sementara pengendalian dot matrik LCD dilakukan secara internal oleh kontroler yang sudah terpasang pada modul LCD.

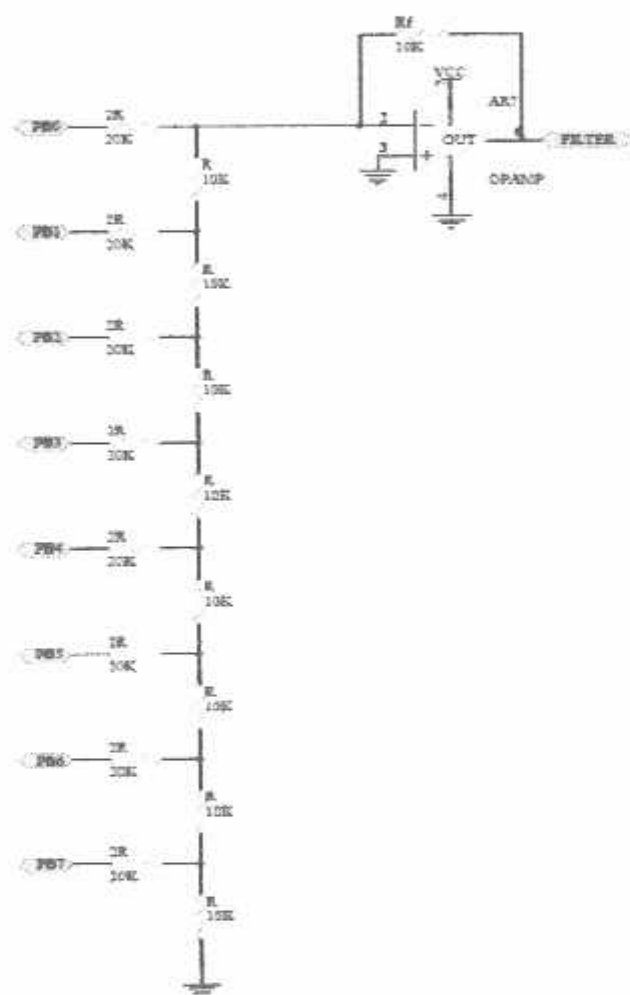
Rangkaian *display* ditunjukkan dalam Gambar 3.7. Saluran data DB₀ – DB₇ dihubungkan pada pin *shift register*. Sedangkan penyemat RS dan Enable dihubungkan pada port 1.0 dan port 1.1 mikrokontroler AT89S8252. Penyemat V_{cc} dihubungkan pada potensiometer 1 K Ω , untuk mengatur kecerahan LCD.



Gambar 3-7 Rangkaian Antarmuka Modul LCD
(Sumber : LCD databook)

3.3.6 Perancangan DAC(Digital/Analog Converter)

Rangkaian DAC(Digital/Analog Converter) berfungsi untuk mengubah sinyal digital dari output mikrokontroller menjadi *output* analog berupa tegangan.DAC R-2R bahwa tegangan *output* analog dapat diatur seperti pada gambar 3-8 merupakan rangkaian DAC (Digital/Analog Converter)



Gambar 3.8
Rangkaian DAC(Digital/Analog Converter)

Saat *handset* ditutup (*on-hook*) maka sentral telepon dan pelanggan menjadi *loop* terbuka. Pada keadaan ini tidak ada arus listrik yang mengalir sehingga LED pada *optocoupler* yang berfungsi untuk mengaktifkan *fototransistor* tidak menyala. *Fototransistor* yang tidak terbias ini mengakibatkan arus listrik tidak mengalir ke *ground* sehingga tegangan pada *port* sama dengan V_{CC} . Keadaan ini oleh Mikrokontroler akan dideteksi sebagai logika tinggi.

Sebaliknya pada saat *handset* diangkat maka sentral dan pelanggan menjadi *loop* tertutup. LED pada *optocoupler* akan menyala dan mengaktifkan *fototransistor*, sehingga arus listrik akan mengalir dari kolektor ke emitor. Akibatnya tegangan pada *port* sama dengan tegangan pada *ground*, hal ini dideteksi oleh Mikrokontroler sebagai keadaan logika rendah.

Dengan menganggap diode sebagai saklar yang diseri dengan sumber tegangan [Malvino, 1995:42], dapat ditentukan nilai tegangan pada resistor R_3 . Sesuai dengan Hukum Kirchoff ke-2 yang menyatakan bahwa jumlah aljabar semua tegangan pada *loop* tertutup adalah nol, maka :

$$V_{loop} = 2V_{diode} + V_{LED} + V_{res} \dots\dots\dots(3.1)$$

Dengan menganggap diode silikon mempunyai tegangan 0,7V dan LED *optocoupler* mempunyai tegangan 1,15V [Motorola, 1995:2], maka :

$$\begin{aligned} V_{res} &= (1,4 - 1,15)V \\ &= 0,25V \end{aligned}$$

Karena LED *optocoupler* mempunyai arus kerja sebesar 10mA [Motorola 1998:2], dengan menggunakan Hukum Ohm diperoleh nilai resistansi R_3 , sebesar:

$$\begin{aligned} R_3 &= \frac{V_{res}}{I} \dots\dots\dots (3.2) \\ &= \frac{0,25V}{0,01A} \\ &= 25\Omega \end{aligned}$$

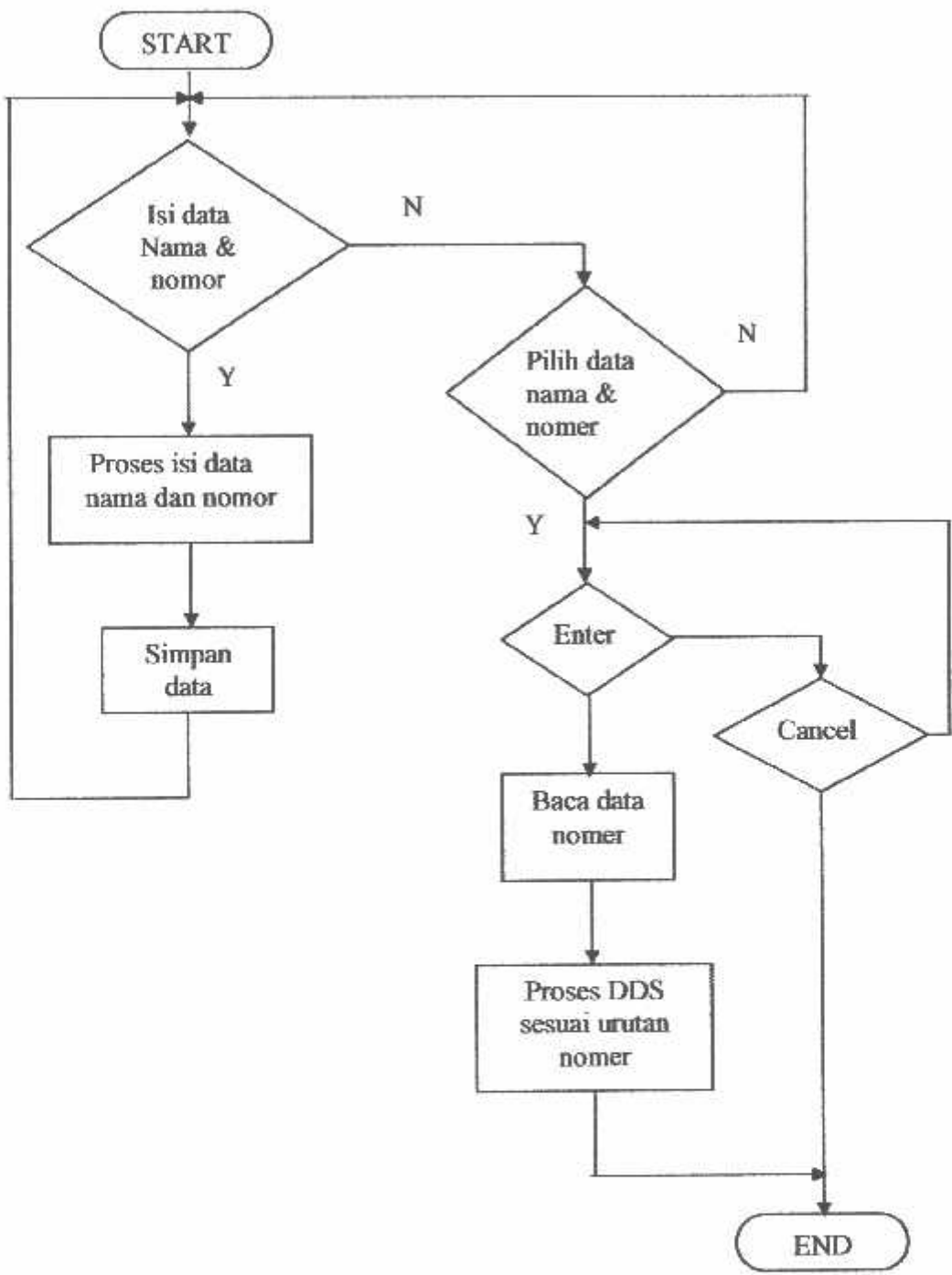
Karena keterbatasan nilai Resistansi resistor yang ada di pasaran maka ditentukan nilai R_3 yang mendekati yaitu 27Ω .

3.3 Perencanaan Perangkat Lunak / *Software*

3.3.1 *Software* di Mikrokontroller (MCU)

Untuk pemakaian mikrokontroler di dalam suatu sistem, perlu direncanakan perangkat lunak mikrokontroler yang dapat mengatur sistem tersebut. Perangkat lunak disini adalah susunan perintah-perintah (program) di dalam memori yang harus dilaksanakan adalah mikrokontroler. Di dalam suatu mikrokontroler memori merupakan suatu fasilitas utama karena disiniilah disimpan perintah-perintah yang harus dijalankan. Memori disini dapat dibedakan menurut fungsinya menjadi memori program dan memori data. *Software* dari alat tersebut terdapat dibagian lampiran dan diagram alirnya adalah seperti terlihat pada gambar 3-9 berikut ini :

3.3.2 Diagram Alir Software Pada Mikrokontroller



Gambar 3-9 Flowchart Sistem

BAB IV

PENGUKURAN DAN PENGUJIAN

4.1. Pendahuluan

Untuk memastikan sistem aplikasi PHONE BOOK digital dan dialer dengan metode DDS pada telepon rumah dapat bekerja sesuai dengan spesifikasi perencanaan, diperlukan serangkaian pengujian dan pengukuran.

Bab pengujian dan pengukuran ini menguraikan tentang bagian alat yang diuji, tujuan pengujian, langkah-langkah pengujian dan hasil pengujian yang menunjukkan unjuk kerja dari tiap-tiap bagian alat. Pembahasan dalam bab ini dibagi menurut pembagian alat yang diuji untuk mengetahui unjuk kerja sistem secara keseluruhan.

Pada bab ini membahas cara pengujian dan analisa dari alat yang dirancang, sehingga dapat diketahui apakah alat tersebut dapat bekerja sesuai dengan yang telah direncanakan. Dalam rangka pengujian alat tersebut diuraikan percobaan yang dilakukan untuk mengetahui respon dari keseluruhan alat yang telah dirancang.

Untuk mengetahui kemampuan alat dan sistem kerja sesuai dengan program yang telah dibuat maka dilakukan pengujian pada alat dan sistem kerja alat dengan prosedur pengujian sebagai berikut:

1. Pengujian perangkat keras
2. Pengujian system secara keseluruhan

4.2 Pengujian Rangkaian DAC R2R

4.2.1 Tujuan

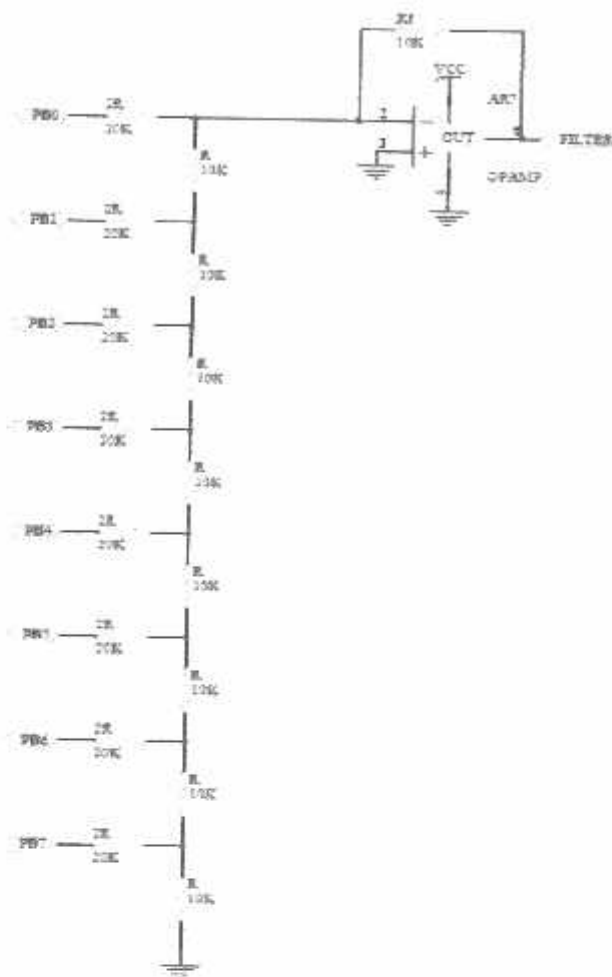
Pengujian rangkaian DAC bertujuan untuk mengetahui apakah DAC berfungsi dengan baik sekaligus untuk mengetahui kelancaran proses konversi atau perubahan sinyal digital ke sinyal analog.

4.2.2 Peralatan yang digunakan

- Rangkaian DAC R2R
- Sumber tegangan +12 volt
- Switch 8 buah
- Multimeter Digital (DT9205B)

4.2.3 Langkah-langkah Pengujian

- 1) Merangkai rangkaian seperti pada gambar dibawah 4-1 :



Gambar 4.1 Rangkaian Pengujian DAC R2R

- 2) Menghubungkan kutub positif multimeter digital (multimeter diset pada skala 20 volt DC) pada outputan DAC, sedangkan kutub negative ke ground.
- 3) Mengamati perubahan nilai tegangan pada multimeter dan mencatat hasilnya pada table 4-1

4.2.4 Analisa

Untuk mengetahui keluaran DAC R2R dapat dicari dengan menggunakan rumus berikut:

$$V_{out} = -\frac{R_f}{R} \times V_{ref} \left[\frac{D_0}{256} + \frac{D_1}{128} + \frac{D_2}{64} + \frac{D_3}{32} + \frac{D_4}{16} + \frac{D_5}{8} + \frac{D_6}{4} + \frac{D_7}{2} \right]$$

Dimana :

$$V_{ref} = 12 \text{ Volt}$$

- Jika diketahui inputan = 00010000_D. Maka berapakah keluaran DAC R2R?

Penyelesaian:

$$V_{out} = -\frac{R_f}{R} \times V_{ref} \left[\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256} \right]$$

$$V_{out} = -\frac{10}{10} \times 12 \left[\frac{0}{2} + \frac{0}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{0}{64} + \frac{0}{128} + \frac{0}{256} \right]$$
$$= -0,75 \text{ Volt}$$

$$\text{Out}_{DAC} = -0.75 \text{ Volt}$$

- Jika diketahui inputan = 11111111_D. Maka berapakah keluaran DAC R2R?

Penyelesaian:

$$V_{out} = -\frac{R_f}{R} \times V_{ref} \left[\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256} \right]$$

$$V_{out} = -\frac{10}{10} \times 12 \left[\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} \right]$$
$$= -11,95 \text{ Volt}$$

$$\text{Out}_{DAC} = -11.95 \text{ Volt}$$

Berikut merupakan table perbandingan tegangan output antara hasil pengukuran dan hasil perhitungan dari keluaran DAC R2R pada table 4-1:

Tabel 4.1

Data Hasil Pengukuran Dan Perhitungan Rangkaian DAC R2R

NO	BIT MASUKAN	KELUARAN DAC R2R TEGANGAN V_0 (Volt)		
		PERHITUNGAN	PENGUKURAN	% KESALAHAN
1	00010000	0,75	0,74	0,132
2	00100000	1,5	1,52	0,131
3	00110000	2,25	2,22	0,135
4	01000000	3	2,90	0,345
5	01010000	3,75	3,61	0,388
6	01100000	4,5	4,38	0,273
7	01110000	5,25	5,17	0,154
8	10000000	6	6,12	0,196
9	11111111	11,95	11,50	0,391

Untuk % kesalahan dapat dihitung dari hasil pengukuran dan perhitungan sebagai

berikut: $\% \text{ kesalahan} = \frac{(\text{perhitungan} - \text{pengukuran})}{\text{perhitungan}} \times 100\%$

Contoh : V_0 perhitungan = -0,75 Volt

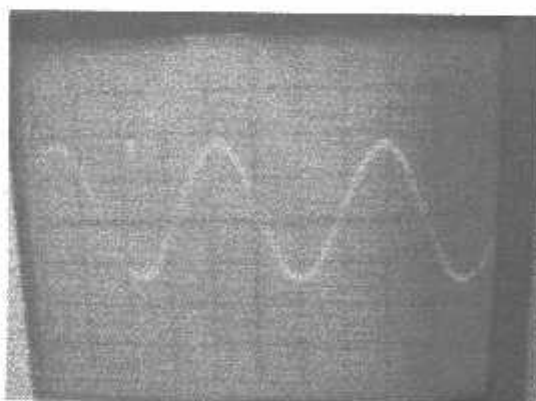
V_0 pengukuran = -0,74 Volt

$$\% \text{ kesalahan} = \frac{(-0,75) - (-0,74)}{(-0,74)} \times 100\% \text{ jadi } \% \text{ kesalahan} = 0,132\%$$



Gambar 4.2 Sinyal Output Sebelum di Filter

Hasil pengujian DAC dapat dilihat dari gambar 4-2, V_{p-p} sebesar 3,6 volt pada Volt/Div sebesar 1 v dan Time/div sebesar 1 ms. Bahwa gambar sinyal yang dihasilkan masih terputus-putus hal ini terjadi karena *input* yang masuk ke DAC berupa data digital sehingga *output* DAC pun masih berupa data 8 bit = 256 langkah. Dari data yang dianalogkan untuk mencapai hasil yang diinginkan maka perlu ditambahkan *filter* pada output rangkaian DAC sehingga akan dihasilkan sinyal *output* yang lebih baik seperti gambar 4-3, V_{p-p} sebesar 3,6 V pada Volt/div sebesar 1 V dan Time/div sebesar 1 ms.



Gambar 4.3 Sinyal Output Setelah di Filter

4.3 Pengujian Rangkaian Op Amp

4.3.1 Tujuan

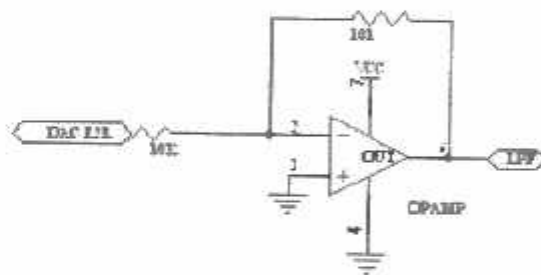
Pengujian rangkaian penguat (Op Amp) bertujuan untuk mengetahui karakteristik penguatan sinyal output sehingga sesuai dengan yang direncanakan.

4.3.2 Peralatan yang digunakan

- Rangkaian Op Amp
- Function Generator
- Voltmeter digital (DT9205B)

4.3.3 Langkah-langkah Pengujian

- 1) Merangkai rangkaian seperti gambar dibawah 4-4 :



Gambar 4-4 Rangkaian Pengujian Op Amp

- 2) Menghubungkan kutub positif multimeter digital (multimeter diset pada skala 10 Volt DC) pada pin 1 dari IC LM741, sedangkan kutub negative ke ground.
- 3) Mengamati perubahan nilai tegangan pada multimeter dan mencatat hasilnya pada table 4-2

Tabel 4-2

Data Hasil Perhitungan dan Pengukuran Rangkaian Op Amp

NO	TEGANGAN INPUT (V)	TEGANGAN OUTPUT (V)		GAIN(A)	% KESALAHAN
		PERHITUNGAN	PENGUKURAN		
1	0.40	-2.5	2.51	6.25	0.4
2	0.42	-2.6	2.62	6.25	0.7
3	0.45	-2.8	2.81	6.25	0.31
4	0.48	-3	3.01	6.25	0.33
5	0.50	-3.1	3.12	6.25	0.64

4.3.4 Analisa

Untuk perhitungan tegangan *output op amp* dapat dicari dengan menggunakan rumus :

$$V_{out} = -V_{in} \frac{R_f}{R_i}$$

Penguatannya (A_{CL}) mempunyai persamaan seperti di bawah ini:

$$A_{CL} = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_i}$$

% Kesalahan dapat dicari dengan persamaan di bawah ini:

$$\% \text{ kesalahan} = \frac{(\text{perhitungan} - \text{pengukuran})}{\text{perhitungan}} \times 100\%$$

Untuk $V_{IN} = 1,5$

Maka tegangan keluaran :

$$V_{out} = -V_{in} \frac{R_f}{R_i}$$

$$V_{out} = -1,5 \frac{R_f}{R_i}$$

$$V_{out} = -1,5V$$

Untuk penguatan (A_{CL}) seperti persamaan di bawah ini:

$$A_{CL} = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_i}$$

$$A_{CL} = \frac{1,5}{1,5}$$

$$A_{CL} = 1 \text{ kali}$$

Kesalahan dalam persen dapat dicari dengan persamaan di bawah ini:

$$\% \text{ kesalahan} = \left| \frac{(\text{perhitungan} - \text{pengukuran})}{\text{perhitungan}} \right| \times 100\%$$

$$\% \text{ kesalahan} = \left| \frac{(-1,5 - (-1,51))}{-1,5} \right| \times 100\%$$

$$\% \text{ kesalahan} = 0,67\%$$

4.4. Pengujian Hardware

Pengujian perangkat keras ini mencakup pengujian rangkaian elektronika pada masing-masing blok maupun blok secara keseluruhan yang telah dirancang dengan menggunakan multimeter digital.

Pengujian LCD

4.4.1. Tujuan

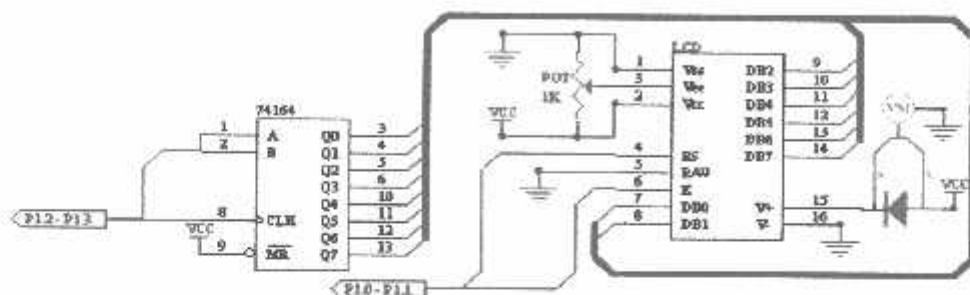
Adapun tujuan dari pengujian rangkaian ini untuk mengetahui kondisi keluaran LCD yaitu sebagai tampilan juga mencatat nilai tegangan yang masuk pada LCD sebelum dan sesudah melewati diode.

4.4.1.1. Alat-alat yang digunakan

- LCD
- Rangkaian mikrokontroler AT 89S8252.
- Catu daya.
- Voltmeter digital

4.4. 1.2. Prosedur pengujian

1. Menyusun rangkaian pengujian seperti pada gambar 4.5



Gambar 4.5 Rangkaian Pengujian LCD

2. Membuat *software* pengujian rangkaian LCD, program ini berisi inisialisasi mikrokontroler dan LCD.
3. Mengaktifkan catu daya.
4. Mengoperasikan program dan hasil keluaran akan ditunjukkan pada layar penampil kristal cair.
5. Mengukur besarnya tegangan awal yang masuk pada LCD dan tegangan setelah lewat pada dioda.

4.4.1.3. Hasil pengujian

Dari hasil pengujian maka didapatkan tampilan seperti yang terlihat pada gambar 4-6;



Gambar 4.6Tampilan Pengujian LCD

Cara menampilkan data diatas adalah dengan memasukan progam seperti dibawah ini :

```
dltone: lcall delay0
```

```
    djnz Dly4,dltone
```

```
    ret
```

```
nama: DB    ' Choiril S '
```

```
nim:  DB    ' NIM: 03.17.030 '
```

```
jurs: DB    ' Teknik Elektro '
```

```
univ: DB    ' ITN Malang  '
```

dan pada waktu pertama dinyalakan maka akan keluar seperti gambar

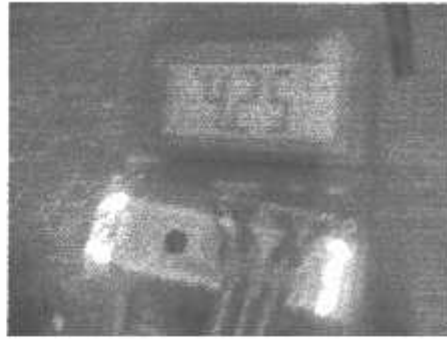
Tabel 4.3 Hasil Pengujian Rangkaian LCD

No	Tegangan Awal LCD (Volt)	Tegangan Setelah Melewati Dioda (Volt)
1	4,87	4,25

(Sumber : pengujian)



Gambar 4.7 Pengukuran Tegangan Awal LCD



Gambar 4.8 Pengukuran Tegangan Setelah Melewati Dioda.

4.4.2. Pengujian Keypad 4 x 4

4.4.2.1. Tujuan

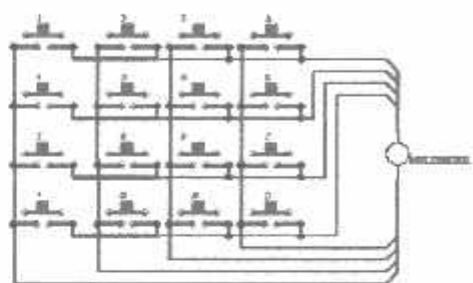
Untuk menguji apakah tombol keypad dapat bekerja sebagai inputan, dan mensimulasikan tombol yang ditekan melalui suara buzzer pada multimeter digital.

4.4.2.2. Alat yang digunakan

- Multimeter digital
- Keypad

4.4.2.3. Prosedur Pengujian

- 1. Menyusun rangkaian pengujian keypad seperti pada gambar 4.9.



Gambar 4.9 Pengujian Rangkaian Keypad.

- 2. Memberikan kombinasi masukan dengan menekan tombol-tombol keypad dengan memasukkan program sebagai berikut:

```
;  
klmpk0: DB '1' ; 2  
klmpk1: DB 'abc2' ; 4  
klmpk2: DB 'def3' ; 4  
klmpk3: DB 'ghi4' ; 4  
klmpk4: DB 'jkl5' ; 4  
klmpk5: DB 'mno6' ; 4  
klmpk6: DB 'pqrs7' ; 5  
klmpk7: DB 'tuv8' ; 4  
klmpk8: DB 'wxyz9' ; 5  
klmpk9: DB '0' ; 2  
;
```


3. Mengamati hasil penckanan *keypad*. Kemudian mencatat hasil pengamatan pada tabel 4.4

4.4.2.4. Hasil Pengujian

Tabel 4.4 Hasil Pengujian *Keypad*.

TOMBOL	BARIS				KOLOM			
	1	2	3	4	1	2	3	4
1	1	0	0	0	1	0	0	0
2	1	0	0	0	0	1	0	0
3	1	0	0	0	0	0	1	0
A	1	0	0	0	0	0	0	1
4	0	1	0	0	1	0	0	0
5	0	1	0	0	0	1	0	0
6	0	1	0	0	0	0	1	0
B	0	1	0	0	0	0	0	1
7	0	0	1	0	1	0	0	0
8	0	0	1	0	0	1	0	0
9	0	0	1	0	0	0	1	0
C	0	0	1	0	0	0	0	1
#	0	0	0	1	1	0	0	0
0	0	0	0	1	0	1	0	0
*	0	0	0	1	0	0	1	0
D	0	0	0	1	0	0	0	1

*Keterangan : cara membaca tabel diatas adalah jika antara baris dan kolom terhubung
(1) maka akan membentuk matrik baris dan kolom sesuai penekanan
tombol *keypad*.



Gambar 4.10 Pengecekan Jalur *Keypad* Dengan Multimeter.

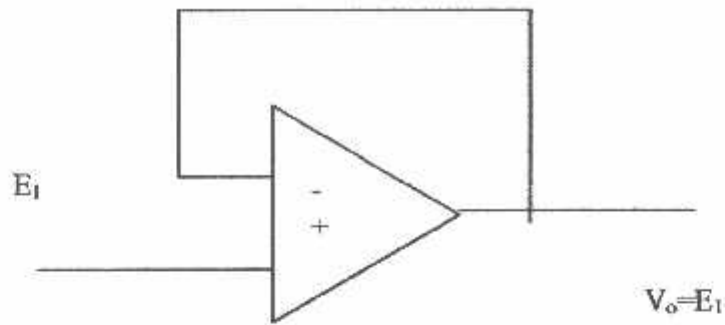
4.4.3. Pengujian Buffer

4.4.3.1. Peralatan Yang Digunakan

- Jumper
- Multi meter digital

4.4.3.2. Langkah Pengujian Buffer

1. Mengukur rangkaian buffer seperti pada gambar 4.11 dibawah ini



Gambar 4.11 Rangkaian Buffer

2. Mengatur Tegangan E_i
3. Mengukur V_{out} dan E_i

4.4.3.3. Analisa

Untuk mengetahui keluaran buffer dapat dicari dengan menggunakan rumus

berikut: $A_{pengukuran} = \frac{V_{out}}{V_{in}}$

Karena pada rangkaian buffer tidak ada R_f dan R_i maka gain sebesar 1x

$$V_{out} \text{ perhitungan} = \left(1 + \frac{R_f}{R_i}\right) + V_{in} = R_f = 0 \, \Omega, R_i = 0 \, \Omega$$

Berikut merupakan table perbandingan tegangan output antara hasil pengukuran dan hasil perhitungan dari keluaran DAC R2R pada table 4-5:

Tabel 4.5

Data Hasil Pengukuran Dan Perhitungan Rangkaian Buffer

	pengukuran	perhitungan	
Vin	Vout	Vout	gain
2	2,01	2	1
4	4,01	4	1
6	6,01	6	1

(Sumber : pengujian)

Jadi

- $V_{in} = 2$

$$V_{out} = 2,01$$

$$\text{Pengukuran} = \frac{2,01}{2} = 1,005$$

4.5. pengujian Sistem Secara Menyeluruh

Pengujian rangkaian secara keseluruhan dilakukan dengan menghubungkan Masing-masing rangkaian atau blok dan menjalankan perangkat lunak yang dibuat. Pengujian ini dimaksudkan untuk mengetahui apakah peralatan yang dibuat telah sesuai. Dengan perencanaan gambar 4-7 merupakan alat secara keseluruhan prosedur pengujian sebagai berikut :

1. pengujian alat digunakan pada daerah yang sama dengan beberapa kali pengukuran.

2. memberi tegangan catu daya 12 Volt DC, dapat berupa baterai, aki dan lain-lain
3. memberi tegangan *buffer* sebesar 5 volt dari microcontroller
4. menghidupkan saklar dengan mengamati LCD
5. mengisi data melalui *keypad*



Gambar 4.12 Alat Secara keseluruhan

Pada pengujian alat diatas pada waktu alat mengeluarkan nada tone yang akan diterima pada mikrophone telephone rumah tidak akan menyambung dikarenakan beberapa noise diantaranya tempat pengujian dekat dengan pemancar radio dan volume suara terlalu besar (suara tone menjadi pecah) maka yang diterima pada mikrophone telephone tidak sama pada standart telkom jadi telephone tidak akan tersambung, apabila daerah tempat pengujian jauh dari pemancar dan volume suara tidak terlalu besar dan tidak terlalu kecil maka telephone rumah akan tersambung pada nomer yang dial tadi

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pengujian dan evaluasi dari perancangan dan pembuatan *PHONE BOOK* digital dan dialer dengan metode *DDS* berbasis mikrokontroler AT89S8252 dikomunikasikan ke telephone rumah melalui nada tone ini maka dapat diambil kesimpulan sebagai berikut:

1. Dari hasil pengujian alat, pada saat microfon telephone rumah didekatkan pada *speaker* yang ada pada *phone book* maka *speaker* akan mengeluarkan nada tone yang dijadikan *out put*, *out put* dari *phone book* dijadikan sandi nomer untuk menghubungi nomer pada saat pengujian
2. Pada pengujian DAC/R2R mempunyai banyak kelebihan dibandingkan ADC yaitu mempunyai setting time yang cepat dan mempunyai resolusi yang halus terhadap frekuensi keluaran dan tegangan output analog yang dapat diatur antara 5V – 34V sesuai dengan yang diinginkan dan satu lagi arus output yang lebih besar dari pada DAC yang lain

5.2. Saran

Semoga pada perancangan dan pembuatan phone book digital dan dialer dengan metode DDS berbasis mikrokontroler AT89S8252 dikomunikasikan ke telephone rumah melalui didekatkan *microfon* telephone rumah dengan *speaker phone book* yang kemudian *speaker* mengeluarkan nada *dual tone* ini nantinya dapat dikembangkan lagi menjadi sebuah sistem dengan tingkatan yang lebih tinggi untuk mengatasi kemajuan teknologi sekarang ini dan phone book digital dan dialer mempunyai kelemahan terhadap antena radio apabila alat dicoba yang tempatnya dekat pemancar radio maka suara radio bisa masuk pada speaker phone book digital maka dari itu bisa dikembangkan lagi yaitu dengan ditambahkan filter untuk menyaring frekuensi dari luar.

DAFTAR PUSTAKA

- [1] Albret paul malvino, hanafi gunawan, prinsip-prinsip Elaktronika,
Erlangga,jakarta,1990.
 - [2] <http://www.atmel.com>
 - [3] <http://www.analog.com>
 - [4] <http://www.Delta-electronic.com>
 - [5] Data sheet IC AD 521, (<http://www.analog devices.com>)
 - [6] Data sheet IC TLC, (<http://www.texas instruments.com>)
 - [7] M 1632 LCD unit user's manual, Seiko instrument inc
 - [8] Malvino, *Prinsip-prinsip Elektronika*, Edisi kedua, Penerbit Erlangga, Jakarta,
1999.
 - [9] Wasito. S, *Vademekum Elektronika*, Edisi kedua, Pt Gramedia Pustaka Utama,
Jakarta, 1995.
 - [10] William D. Cooper, *Instrumentasi Elektronika dan teknik pengukuran*,
Erlangga, Edisi ke-2.
 - [11] Aliday & Resnic, Pantur Silabar Ph & Drs. Erwin Sucipto, *Fisika*, Edisi 3
 - [12] Coughlin, Robert F Driscol, alih bahasa Herman Widodo Soemitro,
Penguat Operasional dan Rangkaian Terpadu Lmer, Edisi Ke-2,
Erlangga, Jakarta,1994.
 - [13] www.Wikipedia.com
-

Lampiran-lampiran



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Jl. Raya Karanglo Km 2 MALANG

PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil Ujian Kompreherensip Jenjang Strata Satu (S1) Jurusan Teknik Elektro
Konsentrasi Elektronika yang diselenggarakan pada :

Hari : Selasa
Tanggal : 24 Maret 2009

Telah dilaksanakan perbaikan skripsi oleh saudara :

Nama : Choiril Setiawan
NIM : 03.17.030

Perbaikan tersebut meliputi :

Penguji	Materi Perbaikan	Paraf
I	1. Pengujian perblok 2. Kesimpulan	
II	1. Abstrak 2. Pengujian DDS 3. Kesimpulan	

Malang, Maret 2009

Disetujui Oleh

Penguji I

Joseph Dedy Irawan, ST MT.
NIP. : 132315178

Dosen pembimbing I

Ir. F. YUDI LIMPRAPTONO, MT
NIP. Y. 1039500274

Penguji II

Bambang Prio Hartono, ST, MT
NIP. Y : 1038400082

Dosen pembimbing II

IBRAHIM ASHARI, ST, MT
NIP. Y. 1030100358

Mengetahui



INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo Km 2
MALANG

FORM BIMBINGAN SKRIPSI

Nama : CHOIRIL SETTAWAN
NIM : 03.17.030
Masa Bimbingan : 9 November 2008 s/d 9 Mei 2009
Judul : Perancangan dan Pembuatan phone book digital + dialer dengan metode DDS berbasis microcontroller AT89S8252)

NO	Tanggal	Uraian	Paraf Pembimbing
1	28/2 2009	bab I — IV Rona acuan, Teori DTMF, Ksya dlt.	
2	3/3 2009	Kesimpulan diura.	
3		Seusun	
4		fac plot	
5			
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing

Ir. F. YUDI LAMPIRAPTONO, MT
NIP. Y. 1039500274

Form S-4a



INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo Km 2
MALANG

FORM BIMBINGAN SKRIPSI

Nama : CHOIRIL SETIAWAN
NIM : 03.17.030
Masa Bimbingan : 9 November 2008 s/d 9 Mei 2009
Judul : Perancangan dan Pembuatan phone book digital + dialer dengan metode DDS berbasis microcontroller AT89S8252)

NO	Tanggal	Uraian	Paraf Pembimbing
1	3/3 2009	Bab I - IV Revisi benarkan yg dicoret	
2	5/3 2009	Bab I - IV beri intro pd gambar dan tabel	
3	5/3 2009	Beri index/narasumber pd gambar	
4	5/3 2009	Daftar Pustaka Kurang	
5	7/3 2009	Tambahlah Gambar cilat	
6	10/3 2009	Revisi SARAN + kesimpulan CAN	
7			
8			
9			
10			

Malang,
Dosen Pembimbing

IBRAHIM ASHAR SLMT
NIP. Y. 1030100358

Form S-4a

Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
 - SPI Serial Interface for Program Downloading
 - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
- 4V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery From Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



8-bit Microcontroller with 8K Bytes Flash

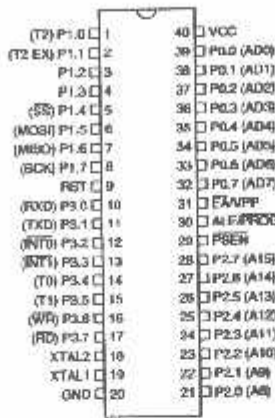
AT89S8252

Rev. 0401E-02/00

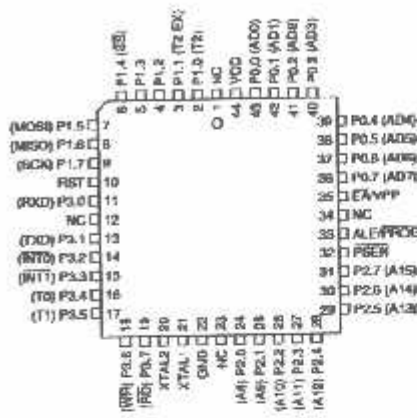


Pin Configurations

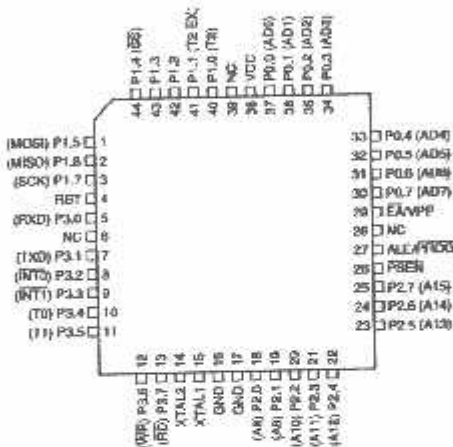
PDIP



PLCC



PQFP/TQFP



Pin Description

VCC

Supply voltage.

ND

Ground.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external

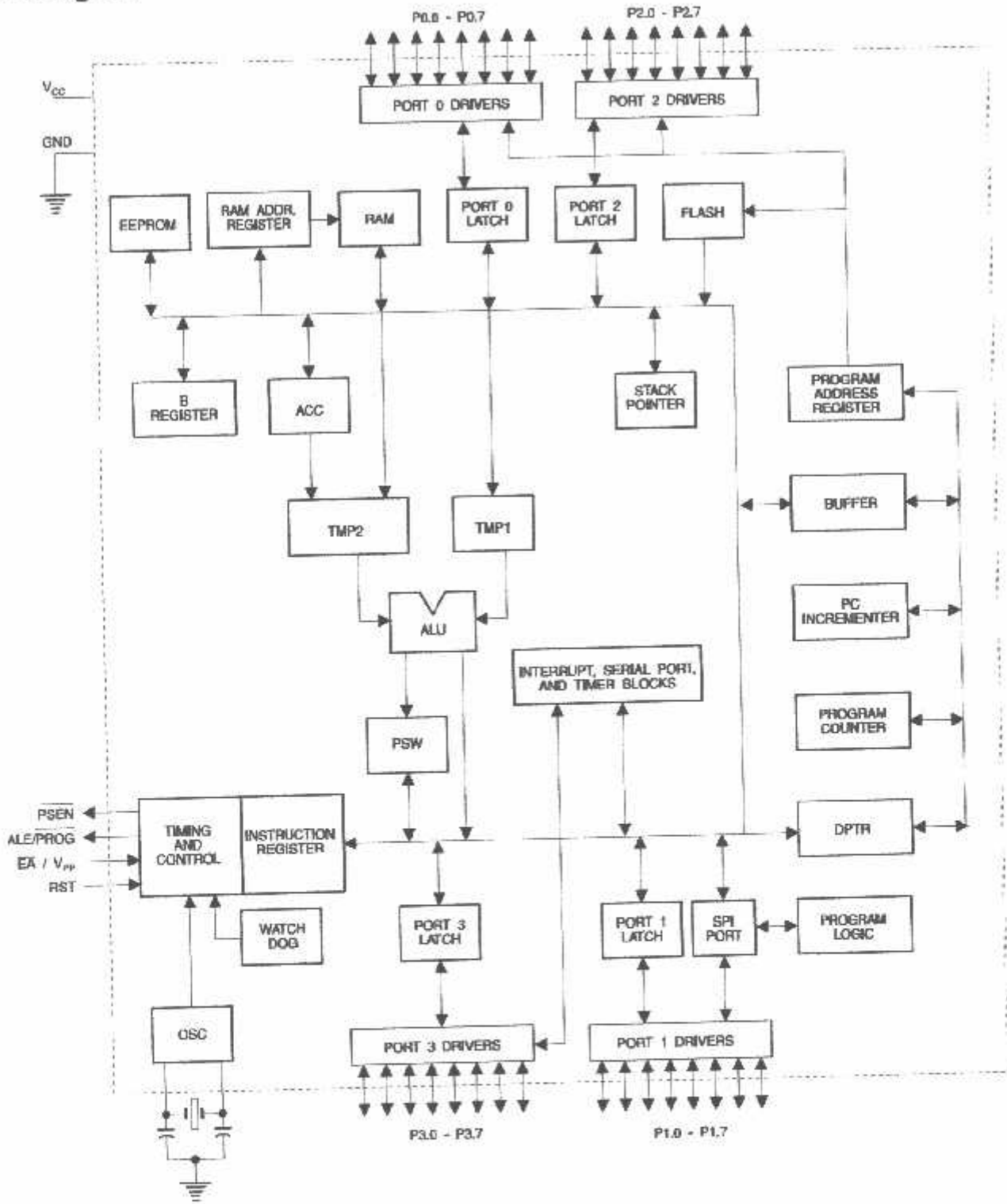
program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Block Diagram



Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Pin Description

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	\overline{SS} (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs,

Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	\overline{RD} (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, \overline{PSEN} is activated twice each machine cycle, except that two \overline{PSEN} activations are skipped during each access to external data memory.

\overline{EA}/VPP

External Access Enable. \overline{EA} must be strapped to GND in order to enable the device to fetch code from external pro-

gram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, \overline{EA} will be internally latched on reset.

\overline{EA} should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S8252 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000					SPCR 000001XX		0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX					0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111						WMCON 00000010	97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000 87H

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted

locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H								Reset Value = 0000 0000B
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used

to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

Table 3. WMCON—Watchdog and Memory Control Register

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H

Reset Value = 0000 0010B

Bit	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
	7	6	5	4	3	2	1	0

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

SPI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by reset.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power Off Flag The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

Table 4. SPCR—SPI Control Register

SPCR Address = D5H

Reset Value = 0000 01XXB

Bit	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
	7	6	5	4	3	2	1	0

Symbol	Function
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects \overline{SS} , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, F_{osc} , is as follows: <div> <div>SPR1SPR0</div> <div>SCK = F_{osc} divided by</div> <div> <div>0 0</div> <div>0 1</div> <div>1 0</div> <div>1 1</div> </div> <div> <div>4</div> <div>16</div> <div>64</div> <div>128</div> </div> </div>

Table 5. SPSR – SPI Status Register

SPSR Address = AAH

Reset Value = 00XX XXXXB

Bit	SPIF	WCOL	—	—	—	—	—	—
	7	6	5	4	3	2	1	0

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

Table 6. SPDR – SPI Data Register

SPDR Address = 86H

Reset Value = unchanged

Bit	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	7	6	5	4	3	2	1	0

Data Memory – EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM or data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 256 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 1.5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the

actual timer periods (at $V_{CC} = 5V$) are within $\pm 30\%$ of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which

he transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

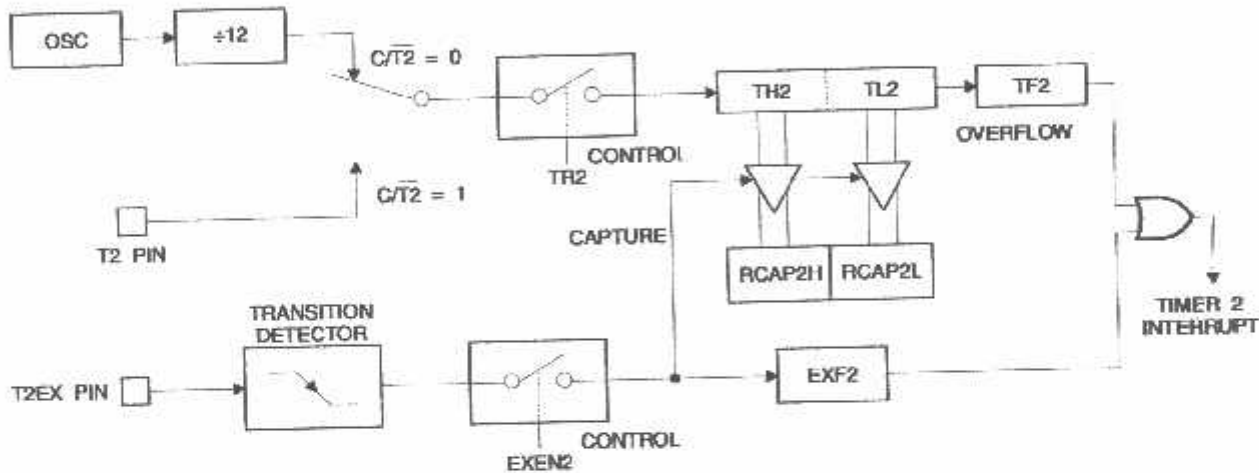
Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Figure 1. Timer 2 in Capture Mode



Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or

by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 in Auto Reload Mode (DCEN = 0)

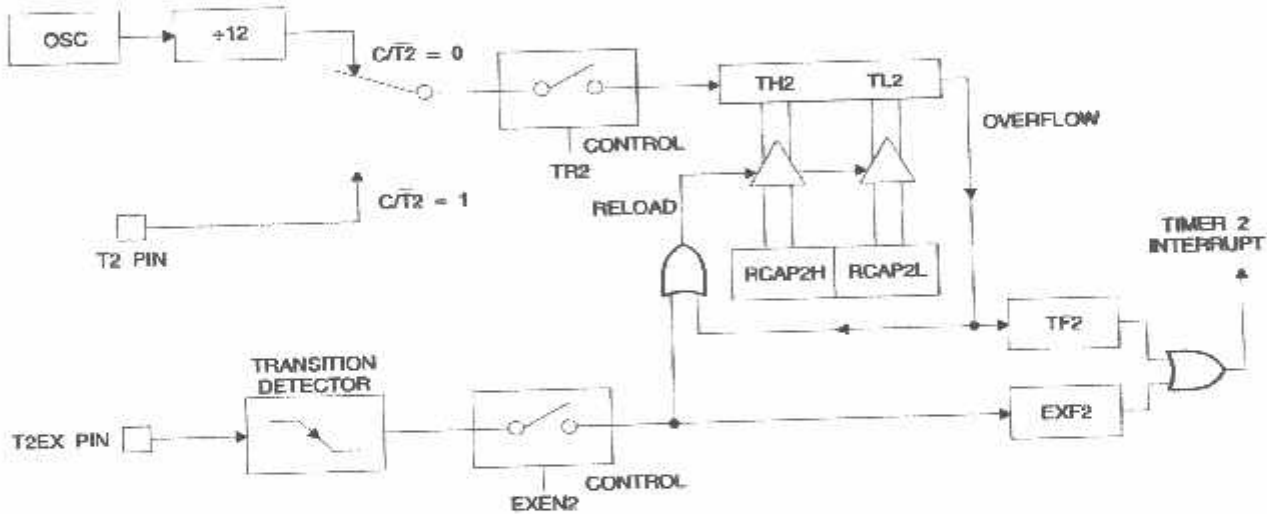


Table 9. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H								Reset Value = XXXX XX00B	
Not Bit Addressable								T2OE	DCEN
Bit	7	6	5	4	3	2	1	0	
Symbol	Function								
–	Not implemented, reserved for future use.								
T2OE	Timer 2 Output Enable bit.								
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.								

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

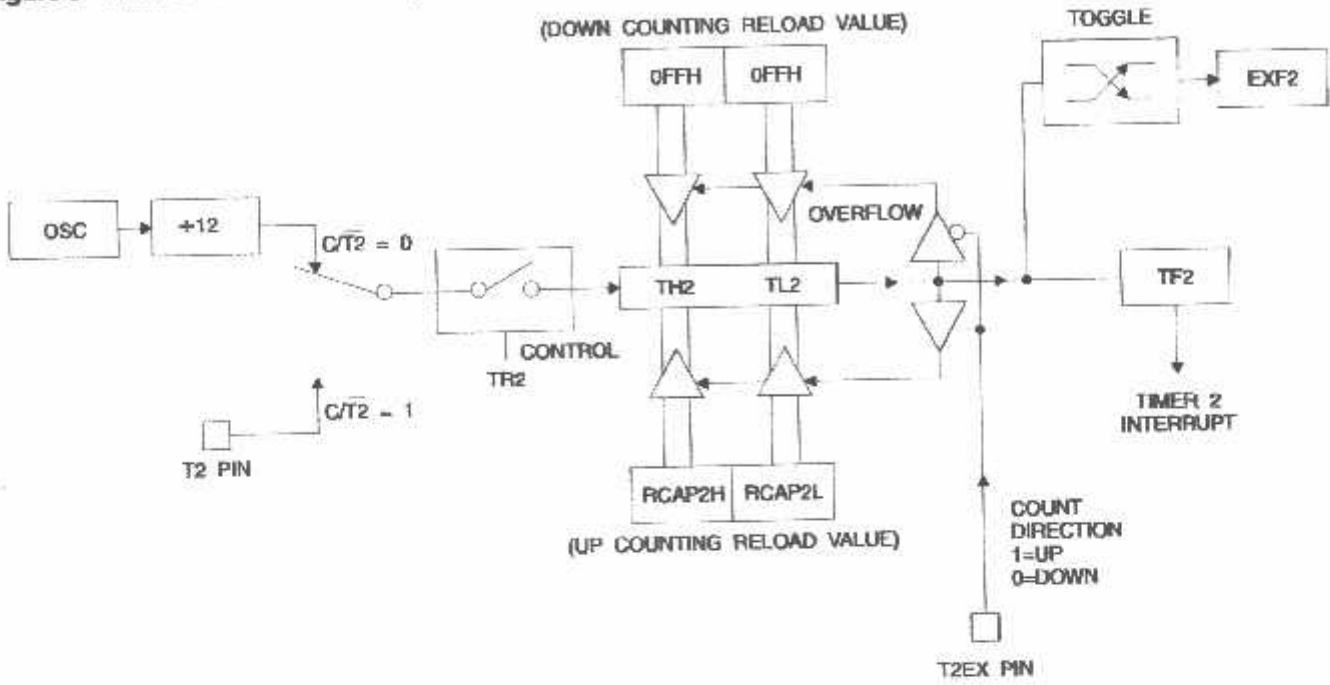
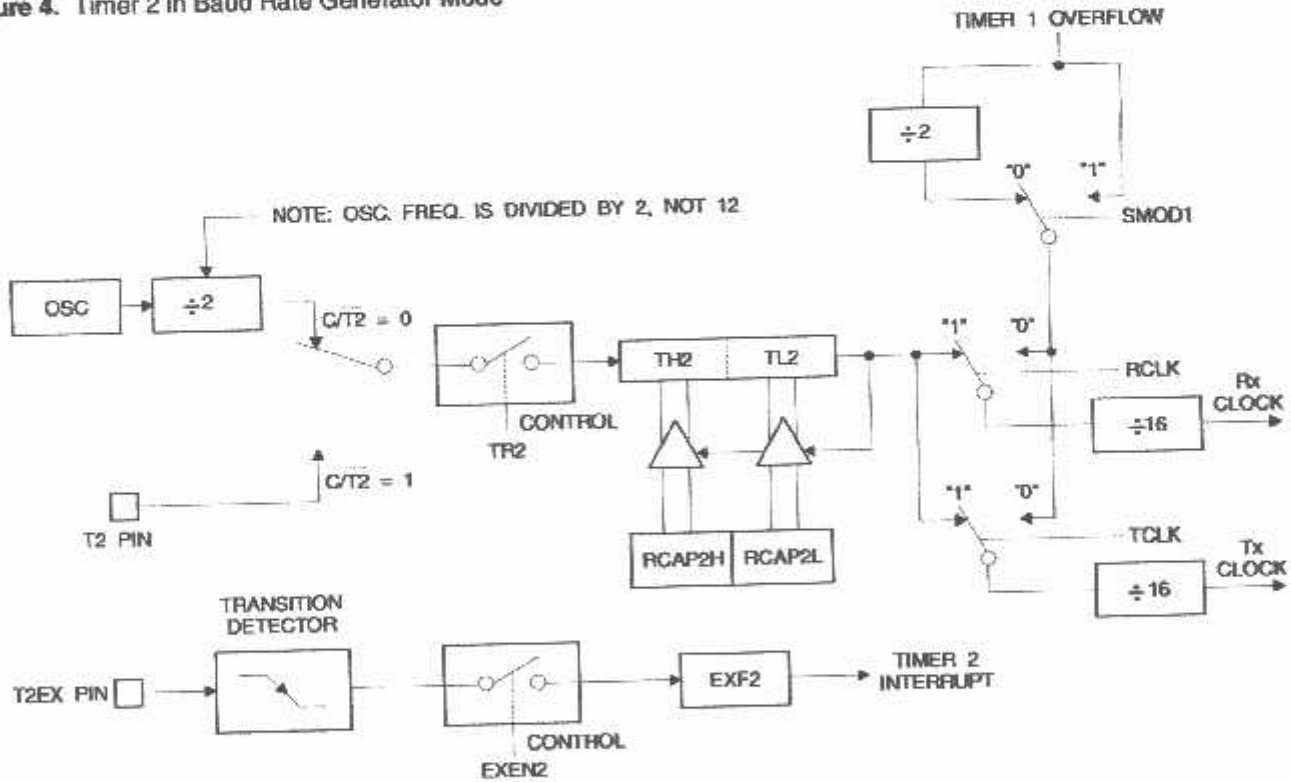


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($CP/T2 = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3 Baud Rate}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer

2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running ($TR2 = 1$) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (RCAP2H, RCAP2L)]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

Figure 5. Timer 2 in Clock-out Mode

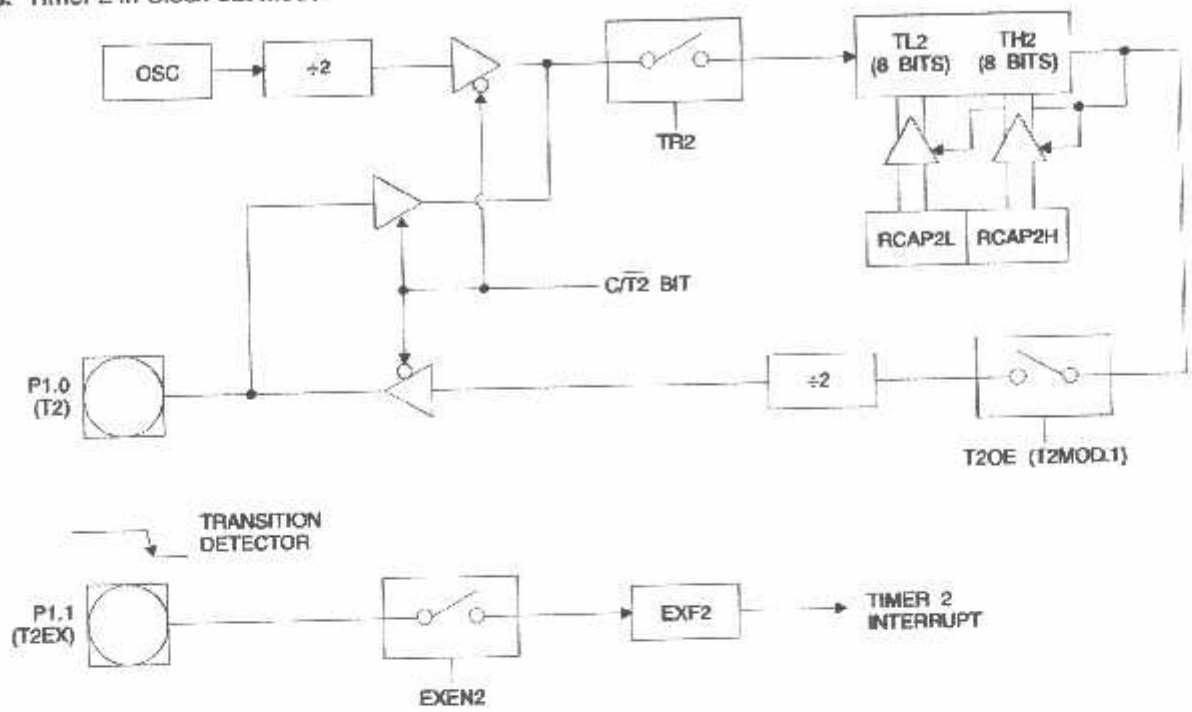
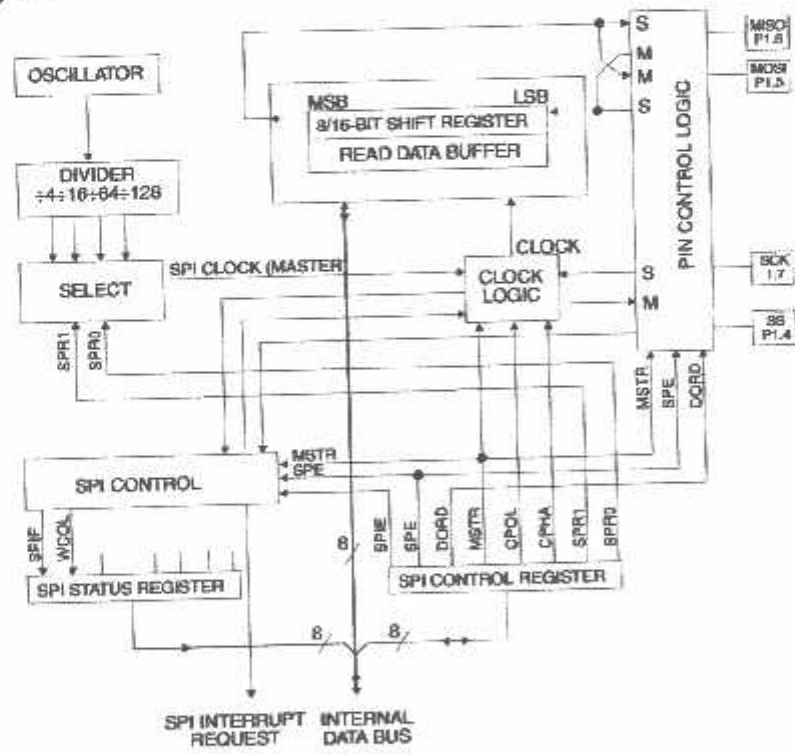


Figure 6. SPI Block Diagram



UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag

- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MISO pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}/P1.4$, is set low to select an individual SPI device as a slave. When $\overline{SS}/P1.4$ is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 8 and Figure 9.

Figure 7. SPI Master-slave Interconnection

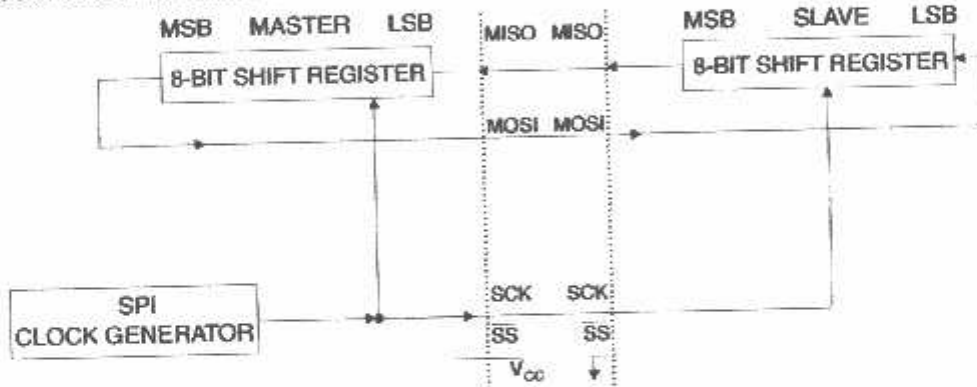
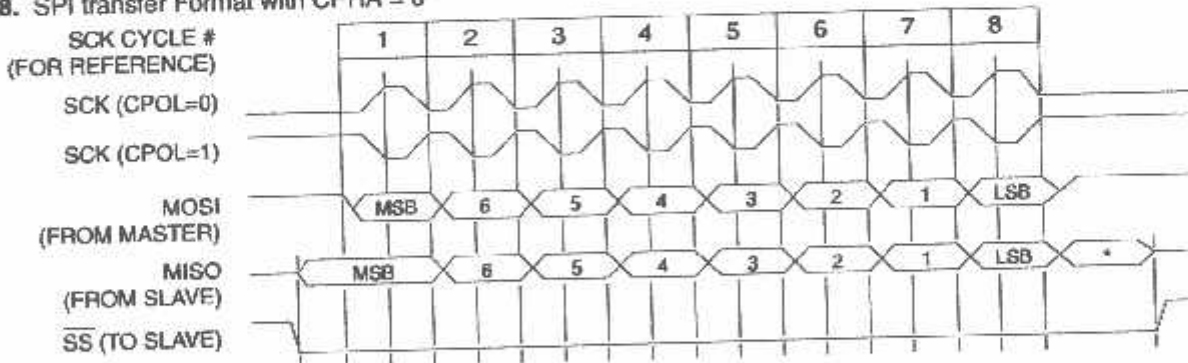
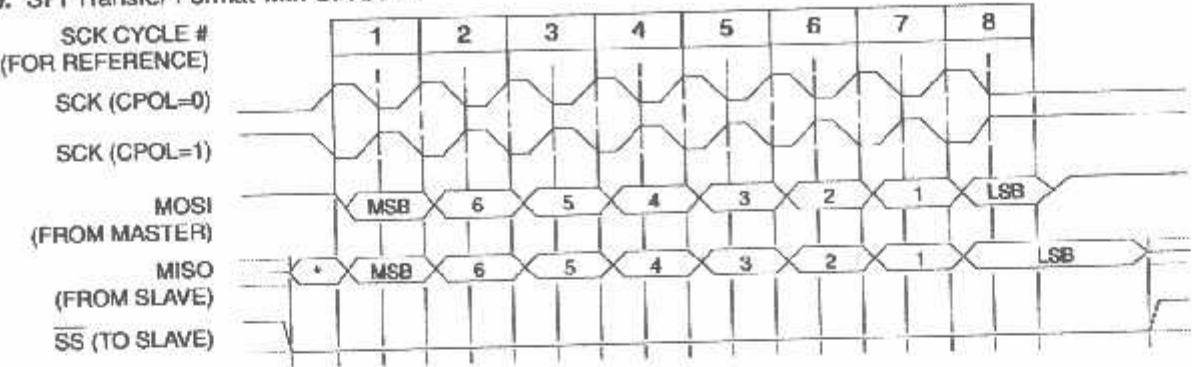


Figure 8. SPI transfer Format with CPHA = 0



Not defined but normally MSB of character just received

Figure 9. SPI Transfer Format with CPHA = 1



Not defined but normally LSB of previously transmitted character

Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S2P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 10. Interrupt Enable (IE) Register

(MSB)(LSB)

EA	—	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

Figure 10. Interrupt Sources

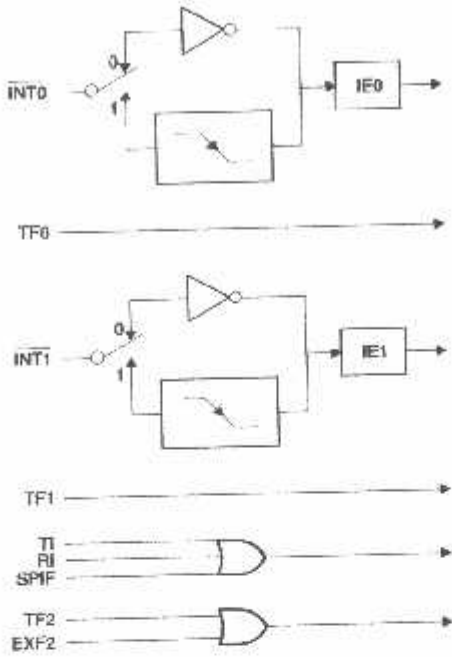
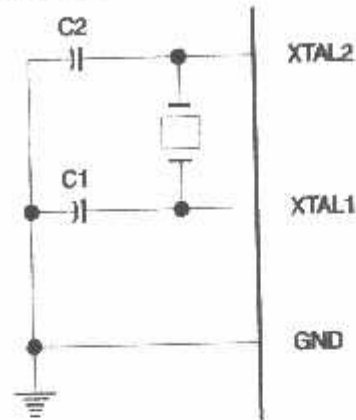
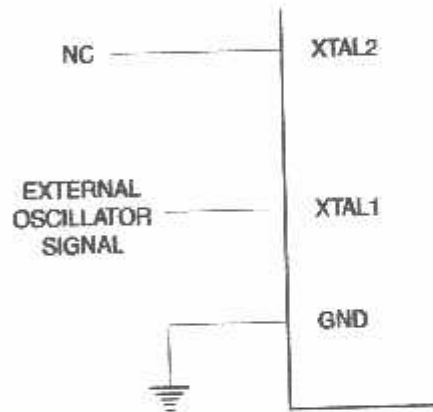


Figure 11. Oscillator Connections



Note: Note: C1, C2 = 30 pF \pm 10 pF for Crystals
= 40 pF \pm 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration



Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution

from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Power-down Mode

In the power-down mode, the oscillator is stopped and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. Exit from power-down can be initiated either by hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power-down via an interrupt, the external interrupt must be enabled as level sensitive before entering power-down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of \overline{EA} must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

Lock Bit Protection Modes⁽¹⁾⁽²⁾

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory. \overline{EA} is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
3	P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as Mode 3, but external execution is also disabled.

Notes: 1. U = Unprogrammed
2. P = Programmed

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-voltage (12V) Parallel programming mode and a Low-voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFFH for the Code array and 2000H to 27FFFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming Algorithm: To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between V_{CC} and GND pins.
 - Set RST pin to "H".
 - Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Set \overline{PSEN} pin to "L".
3. ALE pin to "H".
4. \overline{EA} pin to "H" and all other pins to "H".
5. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
6. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
7. Apply data to pins P0.0 to P0.7 for Write Code operation.

5. Raise \overline{EA}/V_{PP} to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/ \overline{PROG} once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
 - Set XTAL1 to "L".
 - Set RST and \overline{EA} pins to "L".
 - Turn V_{CC} power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Data Polling: The AT89S8252 features \overline{DATA} Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. \overline{DATA} Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate \overline{BUSY} . P3.4 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

Chip Erase: Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/ \overline{PROG} low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.



In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

Serial Programming Fuse: A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT89S8252 is shipped with the Serial Programming mode enabled.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows:

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 72H indicates 89S8252

Programming Interface

Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFFH for Code memory and 000H to 7FFFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between VCC and GND pins.
 - Set RST pin to "H".
 - If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):

- Set XTAL1 to "L" (if a crystal is not used).
- Set RST to "L".
- Turn V_{CC} power off.

Serial Programming Instruction

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:






Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

Note: 1. DATA polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.

2. "aaaaa" = high order address.
3. "x" = don't care.

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{pp}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h ⁽¹⁾	h ⁽¹⁾	x						
Chip Erase	H	L	 ⁽²⁾	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	 ⁽²⁾	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	 ⁽²⁾	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

Notes: 1. "h" = weakly pulled "High" internally.

2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.

3. P3.4 is pulled Low during programming to indicate RDY/BSY.

4. "X" = don't care

Figure 13. Programming the Flash/EEPROM Memory

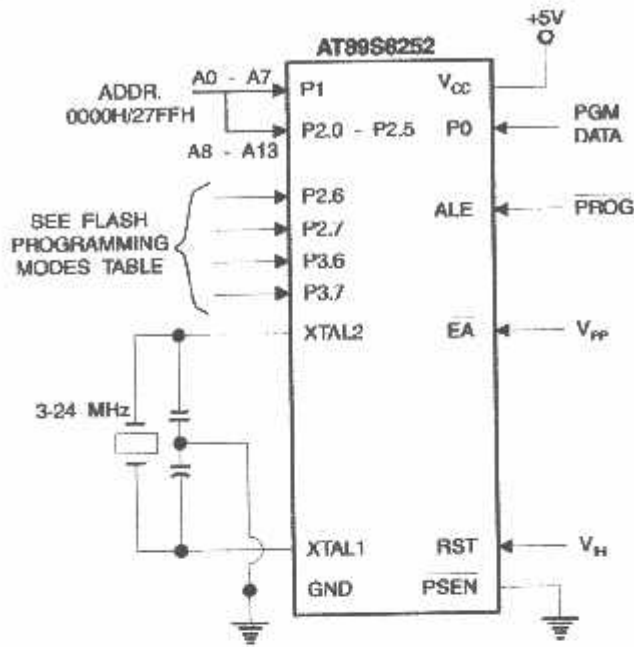


Figure 15. Flash/EEPROM Serial Downloading

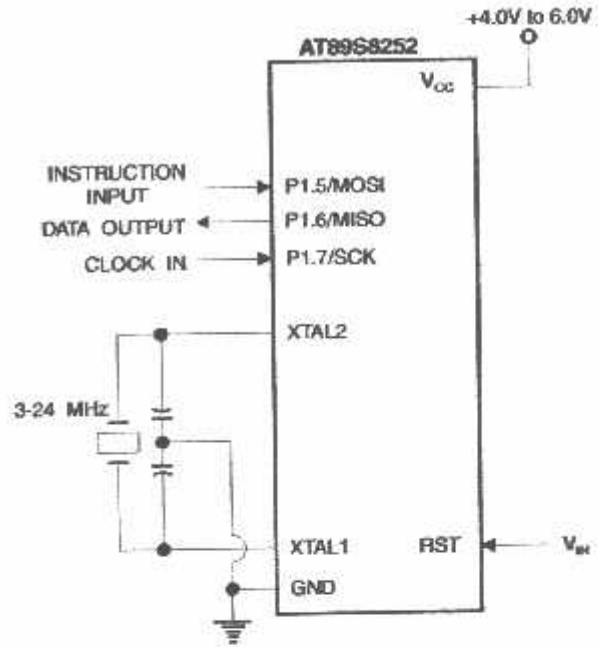
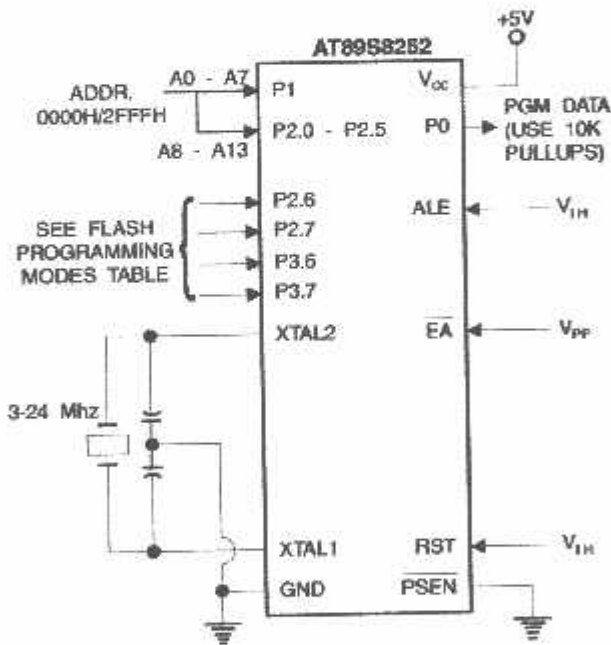


Figure 14. Verifying the Flash/EEPROM Memory

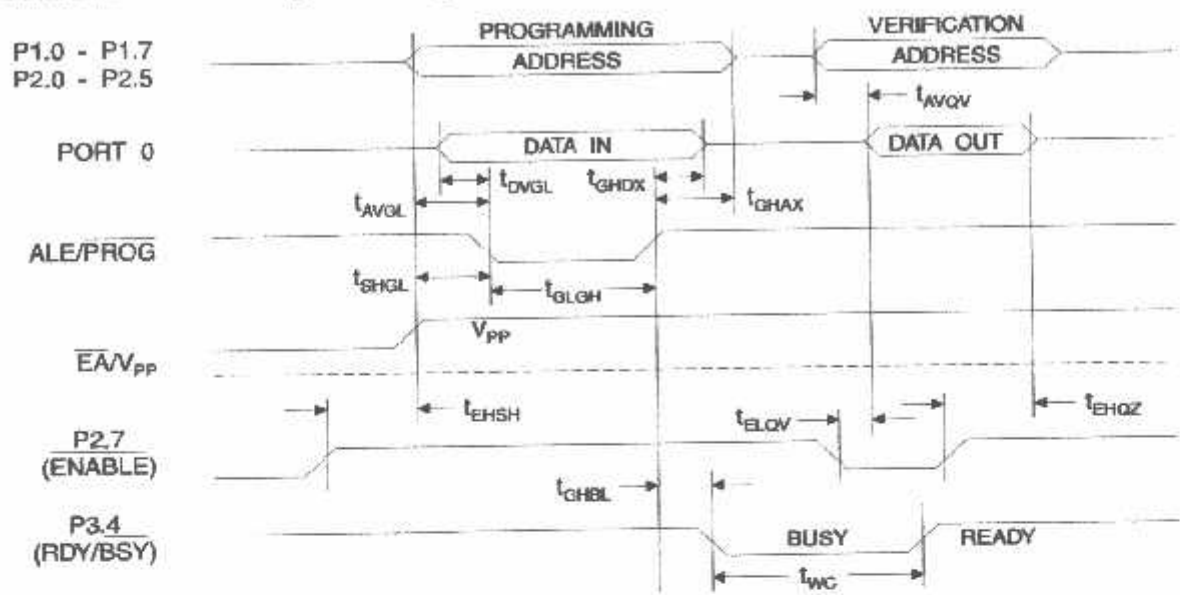


Flash Programming and Verification Characteristics – Parallel Mode

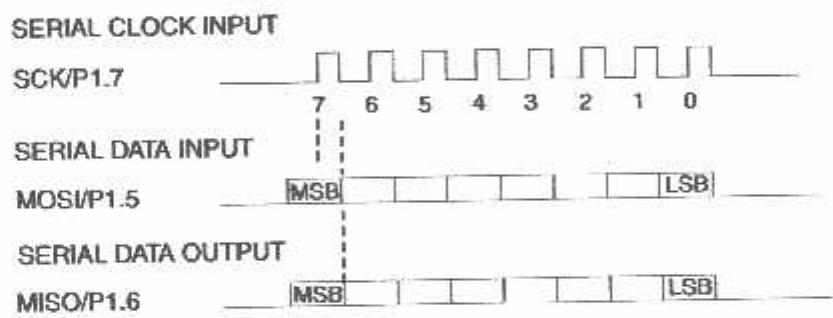
$T_A = 0^\circ\text{C to } 70^\circ\text{C}, V_{CC} = 5.0\text{V} \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{EHSB}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{AVOV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELOV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

Flash/EEPROM Programming and Verification Waveforms – Parallel Mode



Serial Downloading Waveforms



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	5.6V
DC Output Current.....	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 5.0\text{V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low-voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
V_{OL1}	Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.5	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{LI}	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pull-down Resistor		50	300	K Ω
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA
Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V

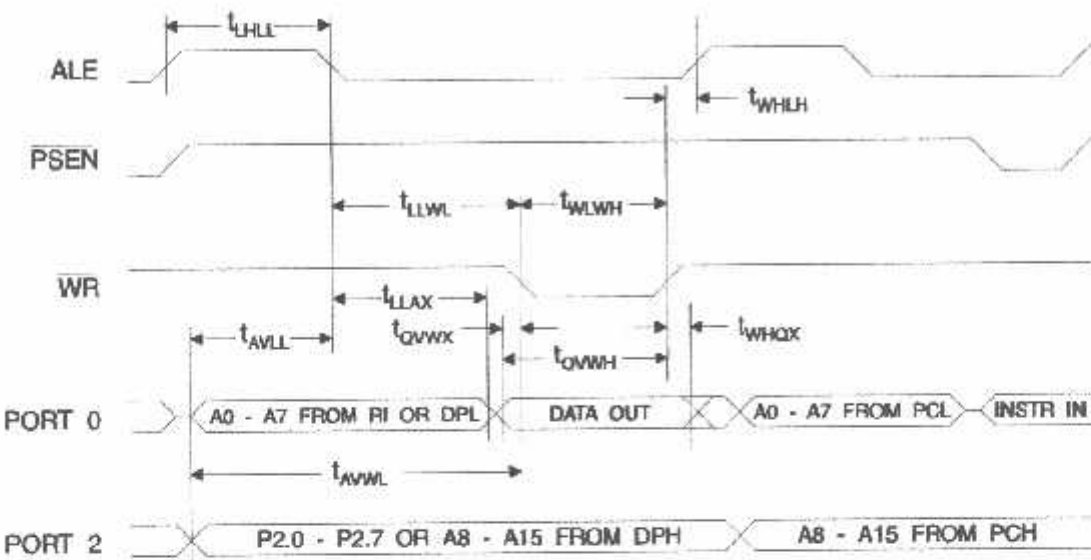
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other outputs = 80 pF.

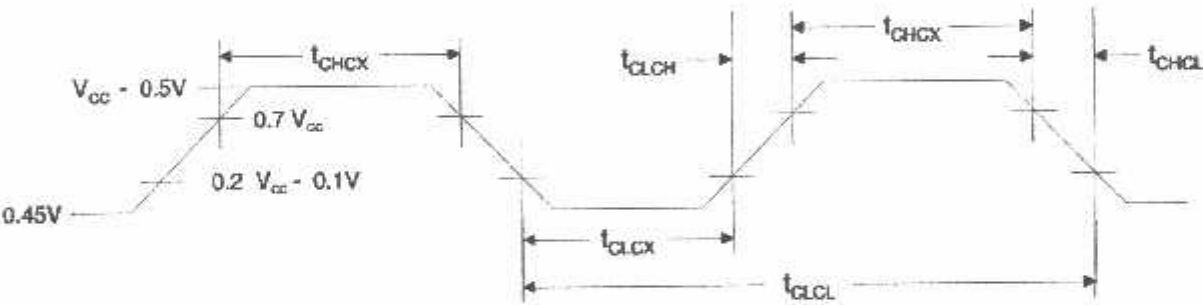
External Program and Data Memory Characteristics

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{AHL}	ALE Pulse Width	$2t_{CLCL} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	$t_{CLCL} - 13$		ns
t_{LLAX}	Address Hold after ALE Low	$t_{CLCL} - 20$		ns
t_{LLV}	ALE Low to Valid Instruction In		$4t_{CLCL} - 65$	ns
t_{LLPL}	ALE Low to PSEN Low	$t_{CLCL} - 13$		ns
t_{PLPH}	PSEN Pulse Width	$3t_{CLCL} - 20$		ns
t_{PLIV}	PSEN Low to Valid Instruction In		$3t_{CLCL} - 45$	ns
t_{PXIX}	Input Instruction Hold after PSEN	0		ns
t_{PXIZ}	Input Instruction Float after PSEN		$t_{CLCL} - 10$	ns
t_{PXAV}	PSEN to Address Valid	$t_{CLCL} - 8$		ns
t_{AVIV}	Address to Valid Instruction In		$5t_{CLCL} - 55$	ns
t_{PLAZ}	PSEN Low to Address Float		10	ns
t_{RLPH}	RD Pulse Width	$6t_{CLCL} - 100$		ns
t_{WLWH}	WR Pulse Width	$6t_{CLCL} - 100$		ns
t_{RLDV}	RD Low to Valid Data In		$5t_{CLCL} - 90$	ns
t_{RHOX}	Data Hold after RD	0		ns
t_{RHOZ}	Data Float after RD		$2t_{CLCL} - 28$	ns
t_{LLDV}	ALE Low to Valid Data In		$8t_{CLCL} - 150$	ns
t_{AVDV}	Address to Valid Data In		$9t_{CLCL} - 165$	ns
t_{LLWL}	ALE Low to RD or WR Low	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
t_{AVWL}	Address to RD or WR Low	$4t_{CLCL} - 75$		ns
t_{QVWX}	Data Valid to WR Transition	$t_{CLCL} - 20$		ns
t_{QVWH}	Data Valid to WR High	$7t_{CLCL} - 120$		ns
t_{WHQX}	Data Hold after WR	$t_{CLCL} - 20$		ns
t_{RLAZ}	RD Low to Address Float		0	ns
t_{WLHI}	RD or WR High to ALE High	$t_{CLCL} - 20$	$t_{CLCL} + 25$	ns

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	V _{CC} = 4.0V to 6.0V		Units
		Min	Max	
1/t _{CLCL}	Oscillator Frequency	0	24	MHz
CLCL	Clock Period	41.6		ns
CHCX	High Time	15		ns
CLCX	Low Time	15		ns
CLCH	Rise Time		20	ns
CHCL	Fall Time		20	ns

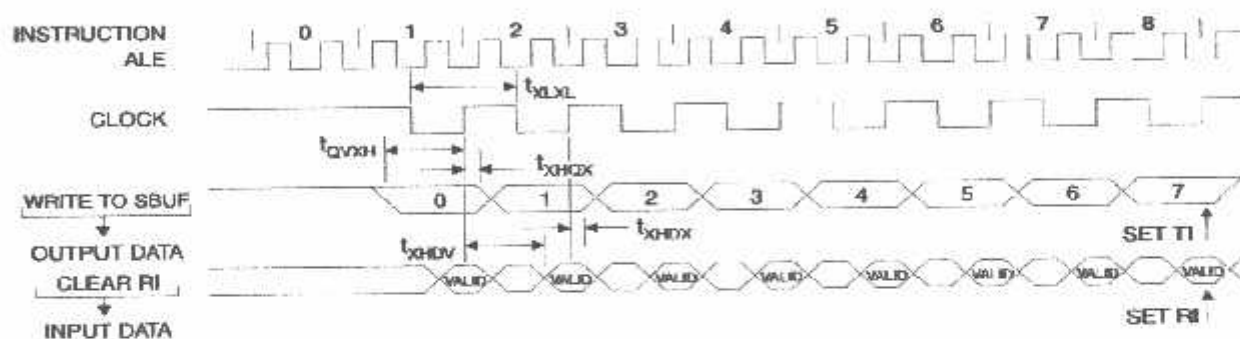


Serial Port Timing: Shift Register Mode Test Conditions

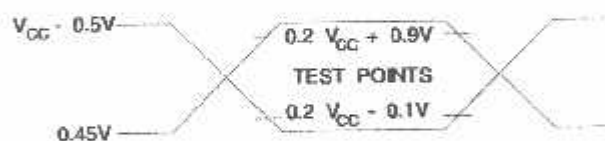
The values in this table are valid for $V_{CC} = 4.0V$ to $6V$ and Load Capacitance = 80 pF .

Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
t_{XCLK}	Serial Port Clock Cycle Time	$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	$10t_{CLCL} - 133$		ns
t_{XHDX}	Output Data Hold after Clock Rising Edge	$2t_{CLCL} - 117$		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		ns
t_{XH0V}	Clock Rising Edge to Input Data Valid		$10t_{CLCL} - 133$	ns

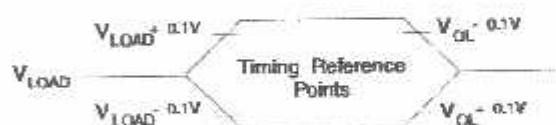
Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾

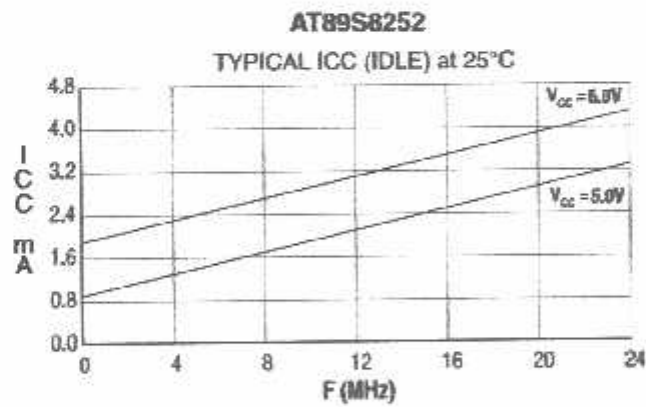
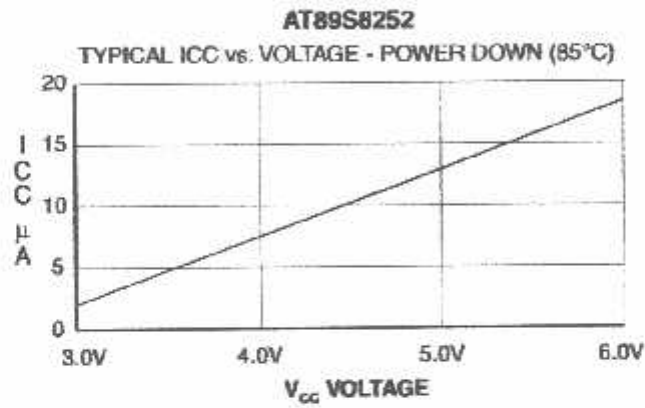
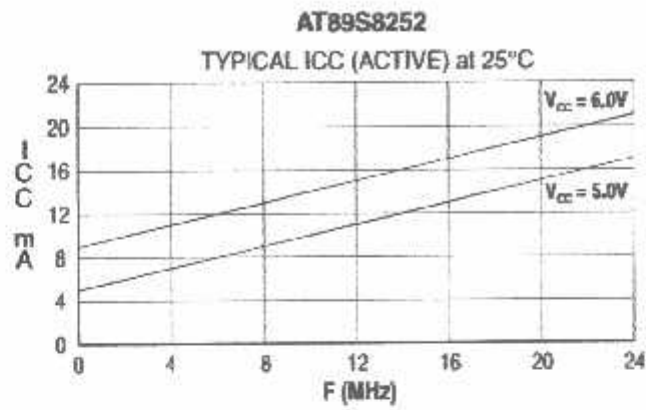


Float Waveforms⁽¹⁾



Notes: 1. AC Inputs during testing are driven at $V_{CC} - 0.5V$ for a logic 1 and $0.45V$ for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Notes: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.



- Notes: 1. XTAL1 tied to GND for I_{cc} (power-down)
2. Lock bits programmed



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 6.0V	AT89S8252-24AC	44A	Commercial (0°C to 70°C)
		AT89S8252-24JC	44J	
		AT89S8252-24PC	40P6	
		AT89S8252-24QC	44Q	
	4.0V to 6.0V	AT89S8252-24AI	44A	Industrial (-40°C to 85°C)
		AT89S8252-24JI	44J	
		AT89S8252-24PI	40P6	
		AT89S8252-24QI	44Q	
33	4.5V to 5.5V	AT89S8252-33AC	44A	Commercial (0°C to 70°C)
		AT89S8252-33JC	44J	
		AT89S8252-33PC	40P6	
		AT89S8252-33QC	44Q	

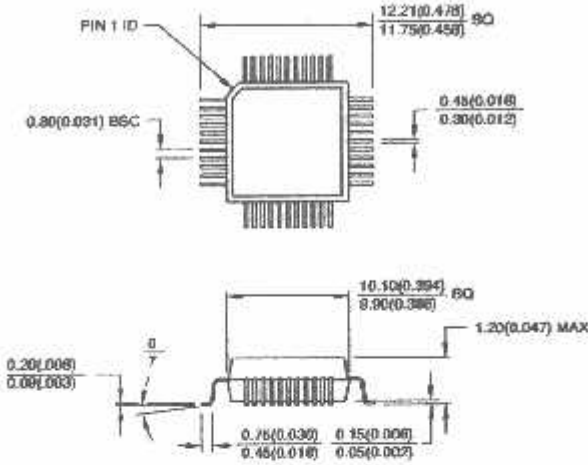
= Preliminary Information

Package Type

4A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
4J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
0P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
4Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

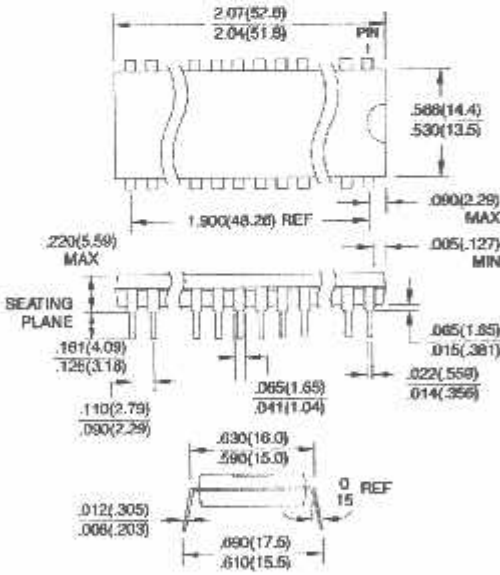
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

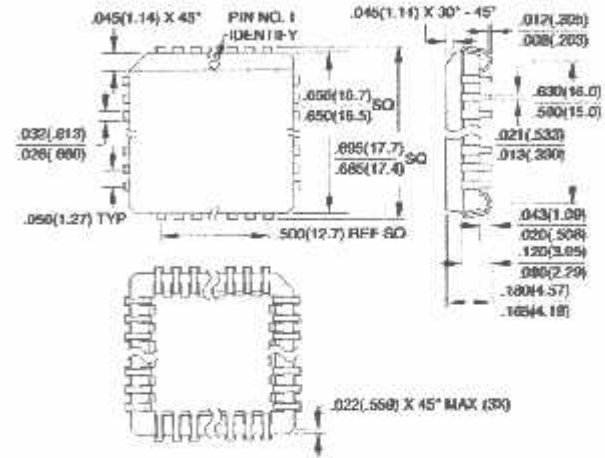


Controlling dimension: millimeters

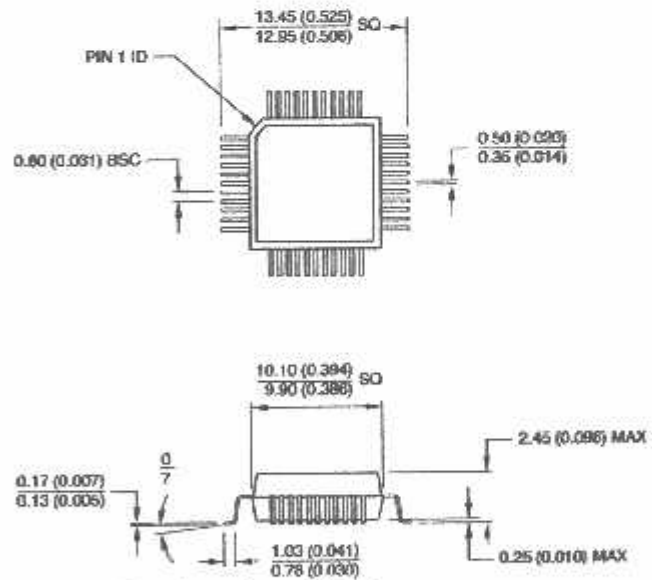
40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs

1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

marks and product names in this document may be trademarks of others.



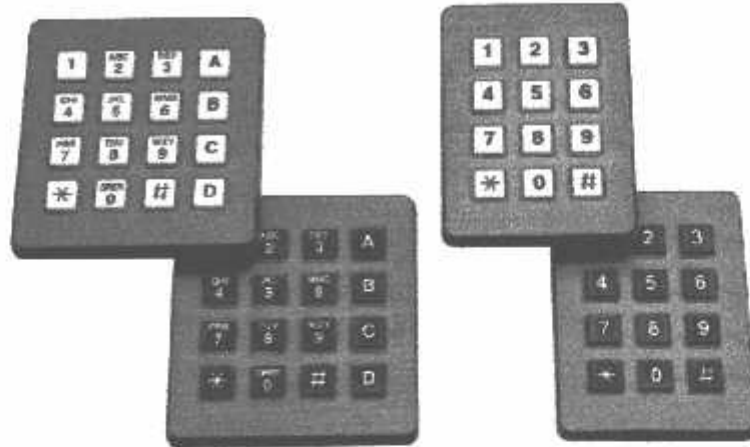
Printed on recycled paper.

0401E-02/00nm

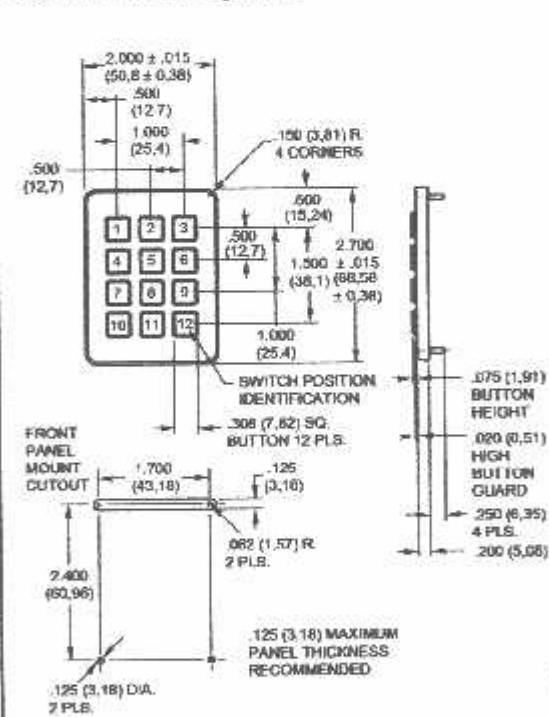
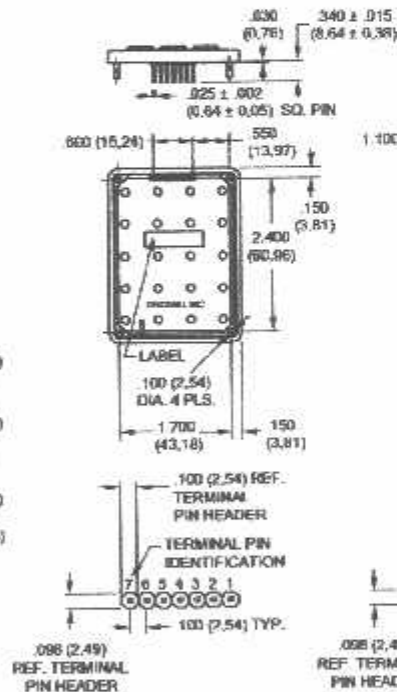
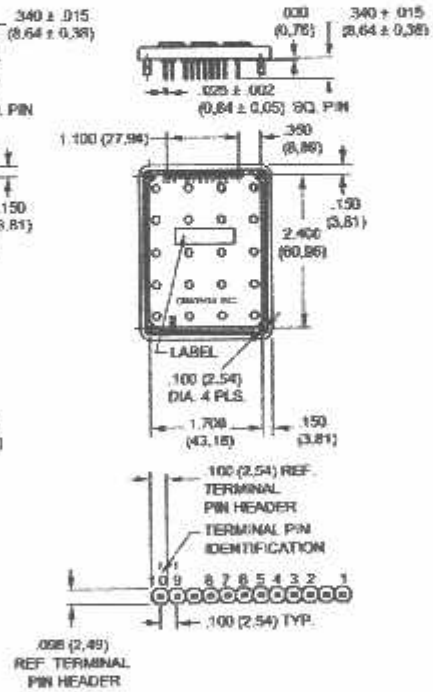
SERIES 96**Conductive Rubber****FEATURES**

- Quality, Economical Keyboards
- Easily Customized Legends
- Matrix Circuitry
- Backlit and Shielded Options Available
- Termination Mates With Standard Connectors
- Tactile Feedback to Operator
- 1,000,000 Operations per Button
- Compatible With High Resistance Logic Inputs

The Series 96 is Grayhill's most economical 3x4 and 4x4 keypad family. The contact system utilizes conductive rubber to mate the appropriate PC board traces. Offered in matrix circuitry, with shielded and backlit options. Built with quality component parts, the Series 96 is subjected to our rigid statistical process control to insure that it meets our reliability standards.

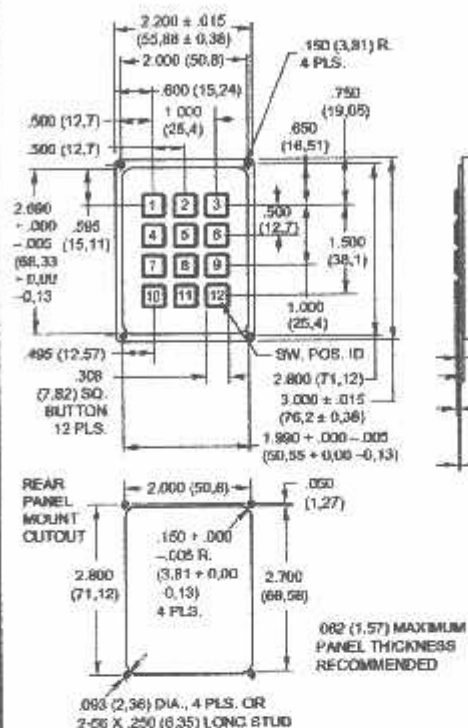


Keypads and Keyboards

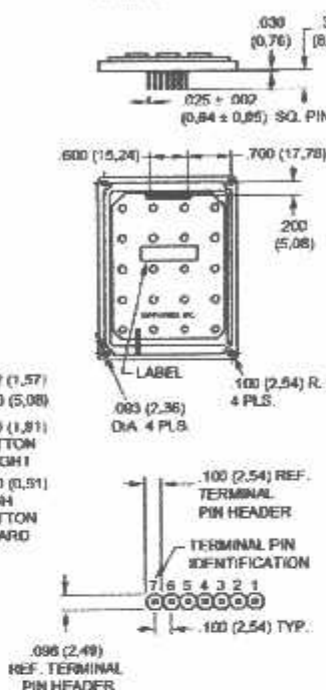
DIMENSIONS In inches (and millimeters)**3x4 Front Mount Keyboard****Standard Versions****Shielded/Backlit Versions**

DIMENSIONS In inches (and millimeters)

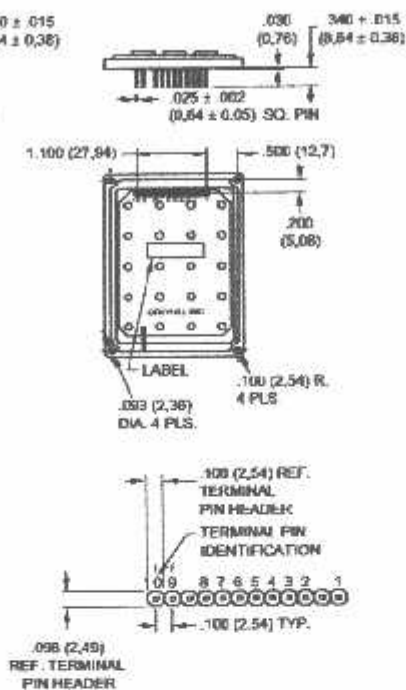
3x4 Rear Mount Keyboard



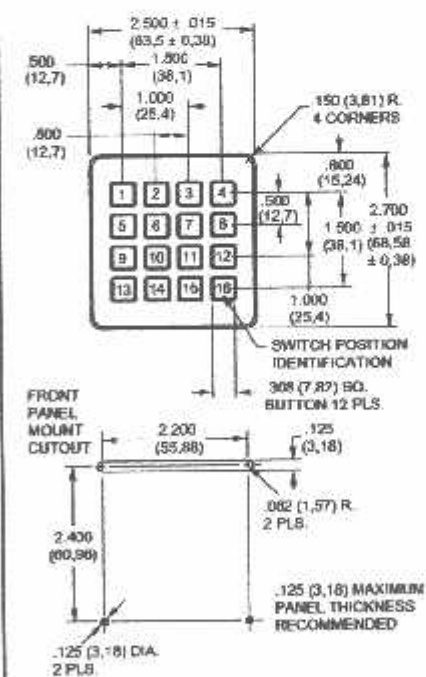
Standard Versions



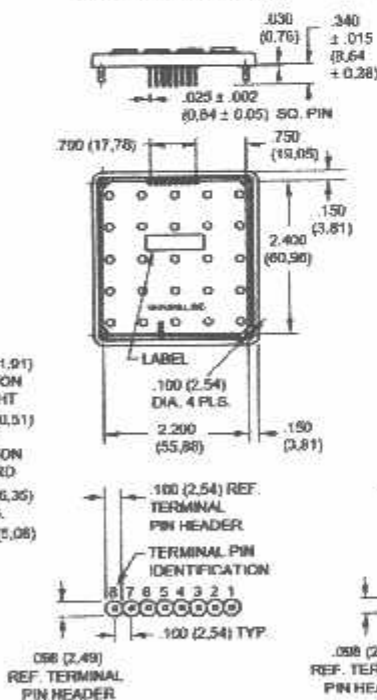
Shielded/Backfit Versions



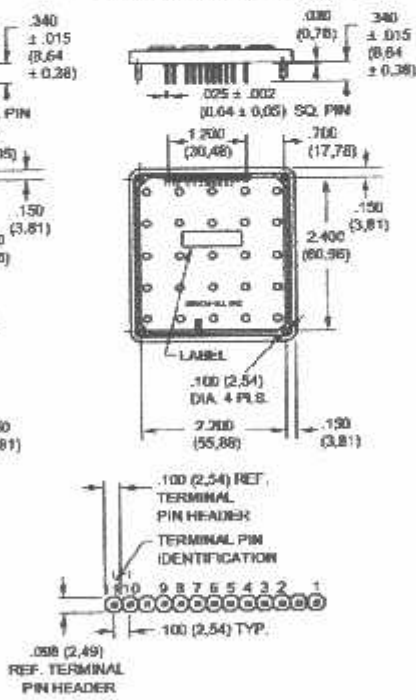
4x4 Front Mount Keyboard



Standard Versions



Shielded/Backlit Versions



SPECIFICATIONS

Rating Criteria

Rating at 12 Vdc: 5 milliamps for .5 seconds

Contact Bounce: < 12 milliseconds

Contact Resistance: < 100 ohms (at stated operating force)

Voltage Breakdown: 250 Vac between components

Mechanical Operation Life: 1,000,000 operations per key

Insulation Resistance: > 10¹² ohms @ 500 Vdc

Push Out Force Per Pin: 5 lbs.

Operating Features

Travel: .040 minimum

Operating Force: 175 ± 40 grams

Operating Temperature: -30°C to +80°C

Material and Finishes

Terminal Pin: Phosphor bronze, solder-plated

PC Board: FR-4 glass cloth epoxy

Keypad: Silicone rubber, durometer 50 ± 5

Housing: ABS, cyclocac "KJW"

Housing Color: Black

Shielding Effectiveness

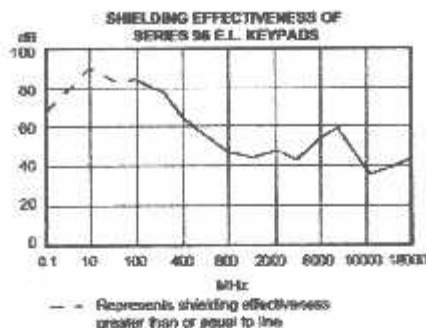
Results shown are typical for a standard Grayhill Series 84S keyboard. A conductive gasket will generally increase the shielding, depending on the size and shape of the gasket and its material. Data derived for E-Field Radiation.

Test Method:

Measurements were made with the keyboard mounted to a brass plate, which in turn was mounted to a shielded enclosure containing the receiving equipment. A signal generator provided the frequency source that was radiated from the transmitting antenna to the enclosed receiving antenna. The spacing between antennas was maintained constant throughout the frequency range. The effectiveness rating is determined by establishing a reference reading without obstruction between the two antennas and determining the difference between that reading and the test setup reading.

Note:

When measured in actual equipment, shielding effectiveness is determined by many factors. This method accurately represents the shielding effectiveness of the Grayhill Series 84S under ideal test conditions.

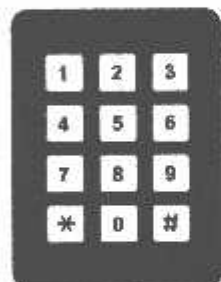


Frequency M Hz	Rating in dB
0.1	≥ 86.2
10	≥ 84.8
100	80.5
400	64.2
800	42.3
2,000	40.5
6,000	33.1
10,000	34.4
18,000	37.0

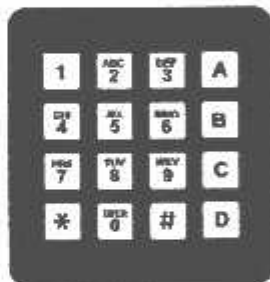
STANDARD LEGENDS

Available through Grayhill Distributors

To order one of the configurations below, use the dash number shown here; select the keypad size and code, and order the part number with the appropriate legend dash number.



-102



-006

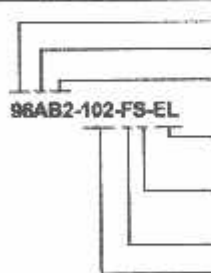


-152



-056

ORDERING INFORMATION



Grayhill Series Number

Keyboard Size: A = 3x4, B = 4x4

Circuitry: B2 = Matrix (terminal pin header)

E.L. Panel Backlighting Option

EL = Backlit, Blank = Non-backlit

EMVRFI Shielding Option

S = Shielded, Blank = Non-shielded

Mounting Option: F = Front panel mount, R = Rear panel mount

Standard Legend Choices

12 Position legends

102 = Black legends on a white button

152 = White legends on a black button

16 Position legends

006 = Black legends on a white button

056 = White legends on a black button

Available from your local Grayhill Distributor. For prices and discounts, contact a local Sales Office, an authorized local Distributor or Grayhill.

M1632 MODULE LCD 16 X 2 BARIS (M1632)

Deskripsi:

M1632 adalah merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LC Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD ini mempunyai CGROM (Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory), DDRAM (Display Data Random Access Memory).

DDRAM

DDRAM adalah merupakan memori tempat karakter yang ditampilkan berada. Contoh, us karakter 'A' atau 41H yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.



Gambar 1
DDRAM M1632 (diambil dari data sheet HD44780)

CGRAM

CGRAM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk dari karakter dapat diubah-ubah sesuai keinginan. Namun memori ini akan hilang saat power supply tidak aktif, sehingga pola karakter akan hilang.

CGROM

CGROM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubahnya. Namun karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif.

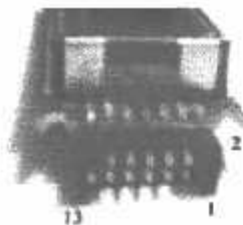
Pada gambar 2, tampak terlihat pola-pola karakter yang tersimpan dalam lokasi-lokasi tertentu dalam CGROM. Pada saat HD44780 akan menampilkan data 41H yang tersimpan pada DDRAM, maka HD44780 akan mengambil data di alamat 41H (0100 0001) yang ada pada CGROM yaitu pola karakter A.

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00000000	00		00P`P							-9Eαp						
00000001	01		!1AQa9							.7チ4āq						
00000010	10		"2BRbr							"イヲxβθ						
00000011	11		#3CScs							」ウチεε						
00000100	00		\$4DTdt							、エトμΩ						
00000101	01		%5EUeu							・オナ16U						
00000110	10		&6FUfu							ヲカニヨρΣ						
00000111	11		'7GWgw							ヲキズラqπ						
00001000	00		(8HXhx							イクヅリJア						
00001001	01)9IYiy							ウツJL"y						
00001010	10		*:JZJz							エコハレJキ						
00001011	11		+;KLk<							オヲヒロ"ス						
00001100	00		,<L#II							オジフワεπ						
00001101	01		-=M]m>							ユズヘンε+						
00001110	10		.>N^n→							ヨセホ"ハ						
00001111	11		/?O_oε							ウツマ"θ						

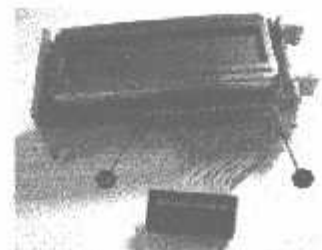
Gambar 2
Hubungan antara CGROM dan DDRAM (diambil dari data sheet HD44780)

Pin Out

No	Nama Pin	FUNGSI
1	VCC	+5V
2	GND	0V
3	VEE	Tegangan Kontras LCD
4	RS	Register Select, 0 = Register Perintah, 1 = Register 1
5	R/W	1 = Read, 0 = Write
6	E	Enable Clock LCD, logika 1 setiap kali pengiriman data
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	Anoda (Kabel coklat untuk LCD Hitachi)	Tegangan positif backlight
16	Katoda (Kabel merah untuk LCD Hitachi)	Tegangan negatif backlight



Gambar 3
Pin Out M1632 LCD Hitachi



Gambar 4
Pin Out LCD M1632 Standard

Register

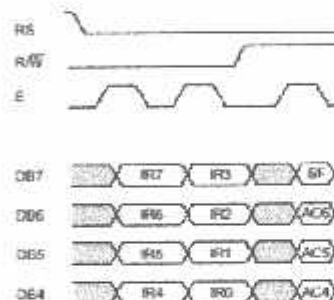
HD44780, mempunyai dua buah Register yang aksesnya diatur dengan menggunakan kaki. Pada saat RS berlogika 0, maka register yang diakses adalah Register Perintah dan pada saat RS berlogika 1, maka register yang diakses adalah Register Data.

Register Perintah

Register ini adalah register di mana perintah-perintah dari mikrokontroler ke HD44780 pada proses penulisan data atau tempat status dari HD44780 dapat dibaca pada saat pembacaan data.

Penulisan Data ke Register Perintah

Penulisan data ke Register Perintah dilakukan dengan tujuan mengatur tampilan LCD, inisialisasi dan mengatur Address Counter maupun Address Data. Gambar 5 menunjukkan proses penulisan data register perintah dengan menggunakan mode 4 bit interface. Kondisi RS berlogika 0 menunjukkan ak data ke Register Perintah. RW berlogika 0 yang menunjukkan proses penulisan data akan dilakukan. Nibble tinggi (bit 7 sampai bit 4) terlebih dahulu dikirimkan dengan diawali pulsa logika 1 pada E Clock. Kemudian Nibble rendah (bit 3 sampai bit 0) dikirimkan dengan diawali pulsa logika 1 pada E Clock. Untuk mode 8 bit interface, proses penulisan dapat langsung dilakukan secara 8 bit (bit 7 ... bit 0) diawali sebuah pulsa logika 1 pada E Clock.



Gambar 5

Timing diagram Penulisan Data ke Register Perintah Mode 4 bit Interface

Tabel 1

Perintah-perintah M1632

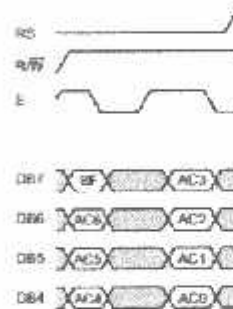
Perintah	I/7	D6	D5	D4	D3	D2	D1	D0	Deskripsi
Display On/Off	0	0	0	0	0	0	0	1	Harus Display dan D0RAAd
Clear Awa	0	0	0	0	0	0	1	X	Set Alamat D0RAAd di 0
Set Mode	0	0	0	0	0	1	1/0	S	Atur arah pengoperan status dan display

Display On/OFF	0	0	0	0	1	D	C	B	Atas display (D) On/OFF, cursor ON/OFF, Blinking (B)
Geser Cursor/Display	0	0	0	1	S/C	R/L	X	X	Geser Cursor atau display target sesuai alamat I/O RAM
Set Fungsi	0	0	1	DL	N	F	X	X	Atas panjang data, jumlah baris, tampilan, dan font karakter
Set CGRAM	0	1	AC0	AC1	AC2	AC3	AC4	AC5	Data dapat dibaca atau ditulis set alamat dasar
Set DDEAM	1	AD0	AD1	AD2	AD3	AD4	AD5	AD6	Data dapat dibaca atau ditulis set alamat dasar

X = diabaikan
 I/D 1-Increment, 0-Decrement
 S 0-Display tidak geser
 S/C 1-Display Shift, 0-Geser Cursor
 R/L 1-Geser Kiri, 0-Geser Kanan
 DL 1-8 bit, 0-4 bit
 N 1-2 baris, 0-1 baris
 F 1-5x10, 0-5x8
 D 0-Display OFF, 1-Display ON
 C 0-Cursor OFF, 1-Cursor ON
 B 0-Blinking OFF, 1-Blinking ON

Pembacaan Data dari Register Perintah

Proses pembacaan data pada register perintah bisa digunakan untuk melihat status busy dari L atau membaca Address Counter. RS diatur pada logika 0 untuk akses ke Register Perintah, R/W diatur pada logika 1 yang menunjukkan proses pembacaan data. 4 bit nibble tinggi dibaca dengan threshold pulsa log 1 pada E Clock dan kemudian 4 bit nibble rendah dibaca dengan diawali pulsa logika 1 pada E Ck. Untuk Mode 8 bit interface, pembacaan 8 bit (nibble tinggi dan rendah) dilakukan sekaligus dengan diawali sebuah pulsa logika 1 pada E Clock.



Gambar 6
 Timing Diagram Pembacaan Register Perintah Mode 4 bit Interface

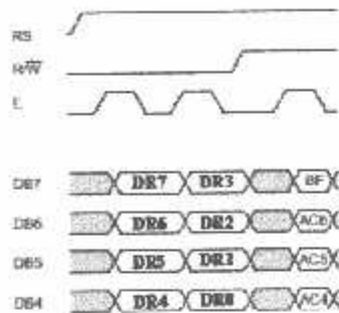
Register Data

Register ini adalah register di mana mikrokontroler dapat menuliskan atau membaca data ke : dari DDRAM. Penulisan data pada register ini akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

Penulisan Data ke Register Data

Penulisan data pada Register Data dilakukan untuk mengirimkan data yang akan ditampilkan pada LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke Register Data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Data 4 bit nibble tinggi (bit 7 hingga

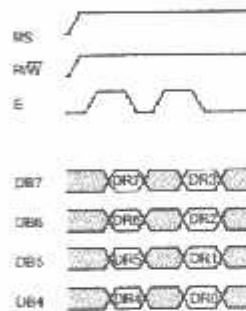
bit 4) dikirim dengan diawali pulsa logika 1 pada sinyal E Clock dan kemudian diikuti 4 bit nibble rendah (bit 3 hingga bit 0) yang juga diawali pulsa logika 1 pada sinyal E Clock.



Gambar 7
Timing Diagram Penulisan Data ke Register Data Mode 4 bit Interface

Pembacaan Data dari Register Data

Pembacaan data dari Register Data dilakukan untuk membaca kembali data yang tampil p LCD. Proses dilakukan dengan mengatur RS pada logika 1 yang menunjukkan adanya akses ke Regi Data. Kondisi R/W diatur pada logika tinggi yang menunjukkan adanya proses pembacaan data. Data 4 nibble tinggi (bit 7 hingga bit 4) dibaca dengan diawali adanya pulsa logika 1 pada E Clock dan dilanjutkan dengan data 4 bit nibble rendah (bit 3 hingga bit 0) yang juga diawali dengan pulsa logika 1 pada E Clock



Gambar 8
Timing Diagram Pembacaan Data dari Register Data Mode 4 bit Interface

Antarmuka LCD dengan mikrokontroler



Gambar 9
Antar muka dengan Modul DST-51



Gambar 10
Antar Muka dengan Modul SC-51 atau AT8951

Program

Rutin-rutin Program untuk DST-51 yang diassembly dengan ALDS atau ASM51

Rutin-rutin Program untuk SC-51/AT8951 yang diassembly dengan ALDS atau ASM51

Rutin delay yang diassembly dengan ALDS atau ASM51

Datasheet HD44780

R2R DIGITAL TO ANALOG CONVERTER

Deskripsi:

DAC - R - 2R Ladder ini memiliki kelebihan-kelebihan antara lain

- Tegangan output analog yang dapat diatur antara 5V sampai 34V sesuai yang diinginkan
- Arus output yang lebih besar daripada IC DAC yang lain.

Hal ini disebabkan DAC ini menggunakan OP-AMP. Secara umum rangkaian DAC

- R - 2R Ladder ini dapat dibagi menjadi 2 bagian utama, yaitu :

1. Blok R-2R Ladder
2. Blok Penguat Tegangan

Blok R-2R Ladder

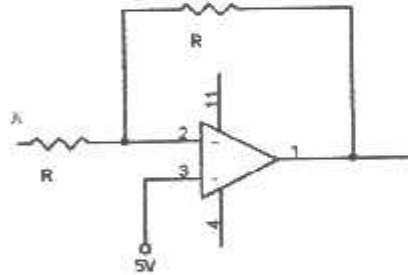
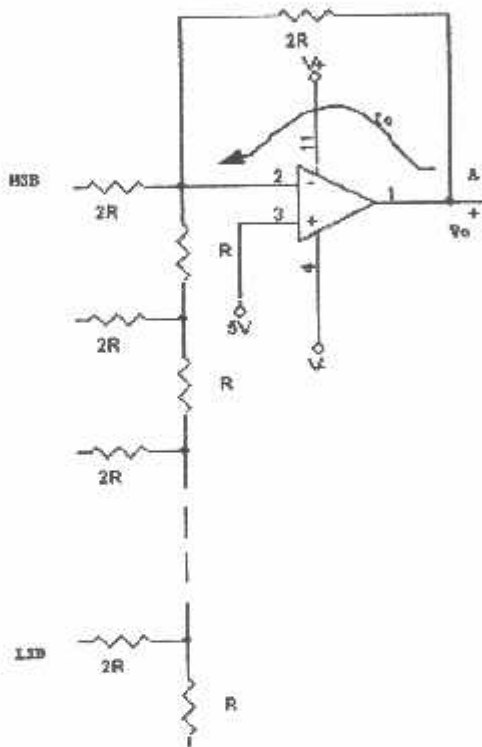
Rangkaian R-2R Ladder mengkonversikan 8 bit data digital dari mikrokontroler menjadi tegangan analog yang bersesuaian

Apabila semua bit data berlogika 1 (FHH) maka tidak ada arus I_o yang mengalir sehingga tegangan output (V_o) = 5V

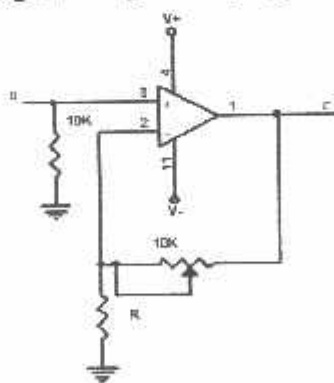
Apabila semua bit data = 0 (0H) maka mengalir arus I_o dan tegangan output = 10V

Pada bagian ini ada suatu level DC 5V yang harus dihilangkan agar output DAC berayun antara 0 sampai 5V.

Untuk itu, output DAC ini diinputkan lagi ke sebuah penguat membalik yang mana akan menghasilkan tegangan output DAC 0-5V saja.



Bagian Penguat Tegangan

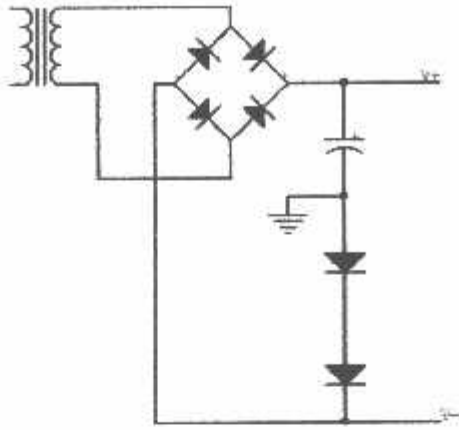


Output DAC ini masih dapat dinaikkan ke level yang lebih besar tergantung dari besar tegangan supply yang digunakan. Tegangan output ini maksimal 1,5V dibawah tegangan supply positifnya. Rangkaian dasar penguat tegangan adalah penguat tegangan tidak membalik seperti dalam gambar berikut

$$V_o = V_c = (1 + R_f/R) V_B$$

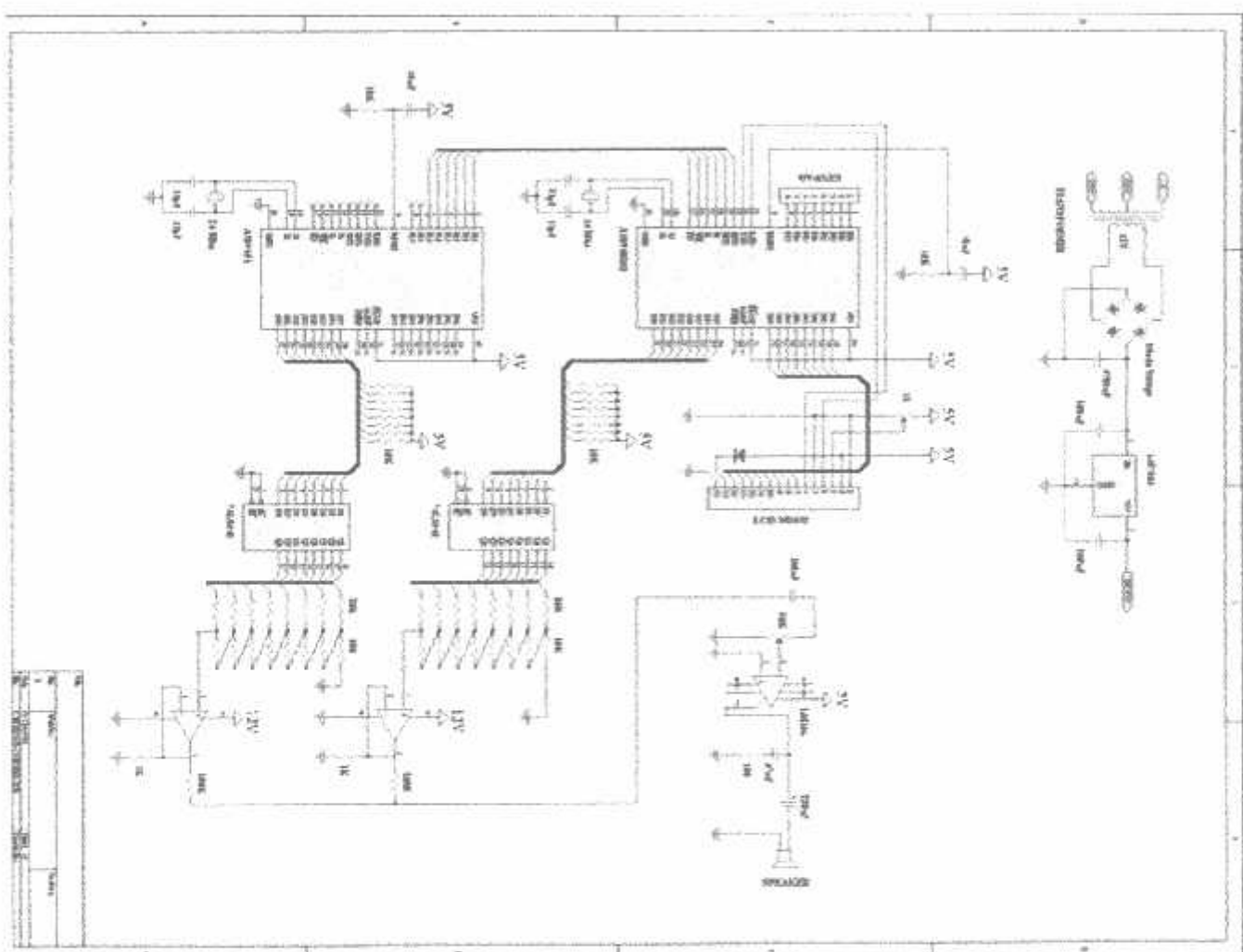


Supply DAC

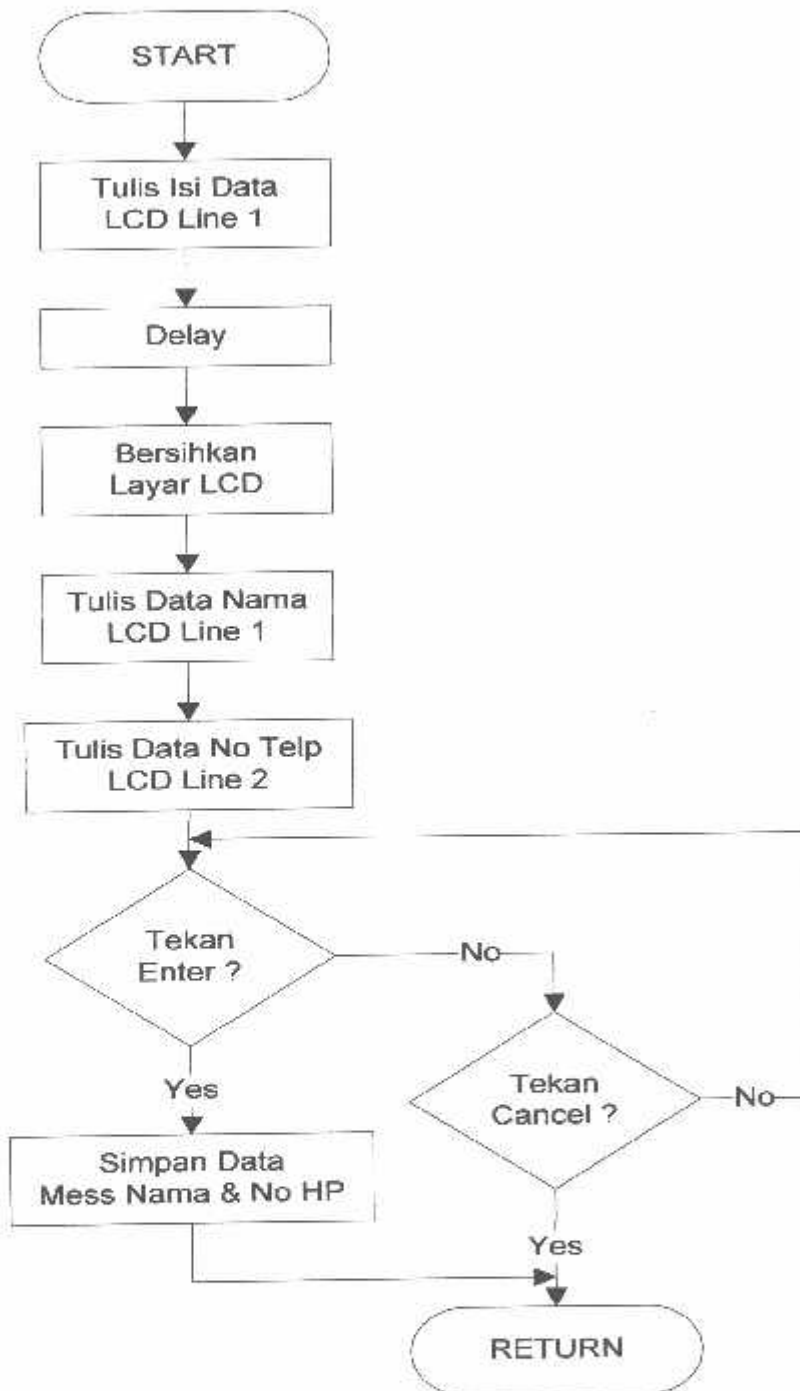


Pada bagian ini, tegangan $-V$ dibutuhkan untuk menghasilkan respon yang lebih linier dari Op Amp yang digunakan.

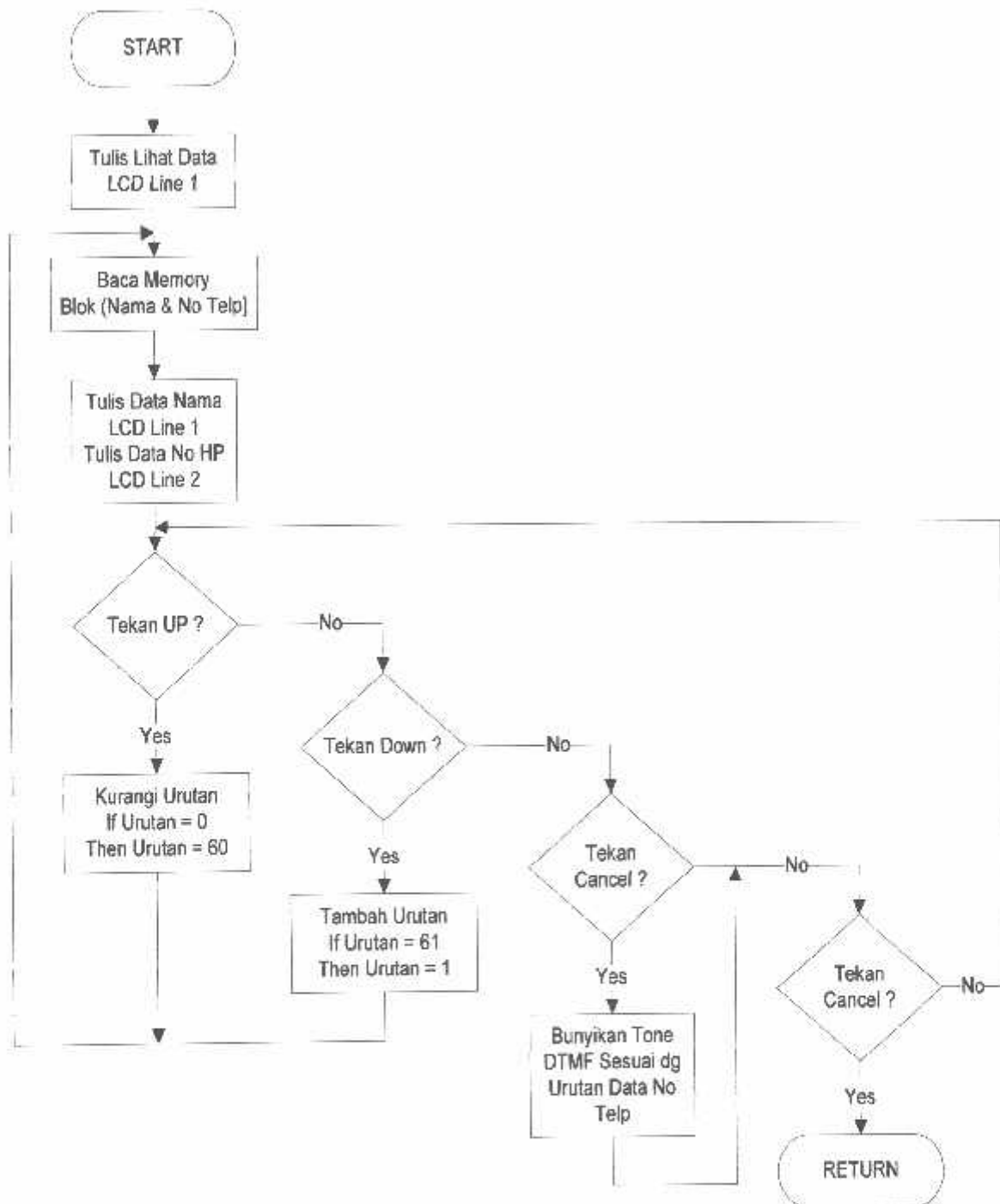


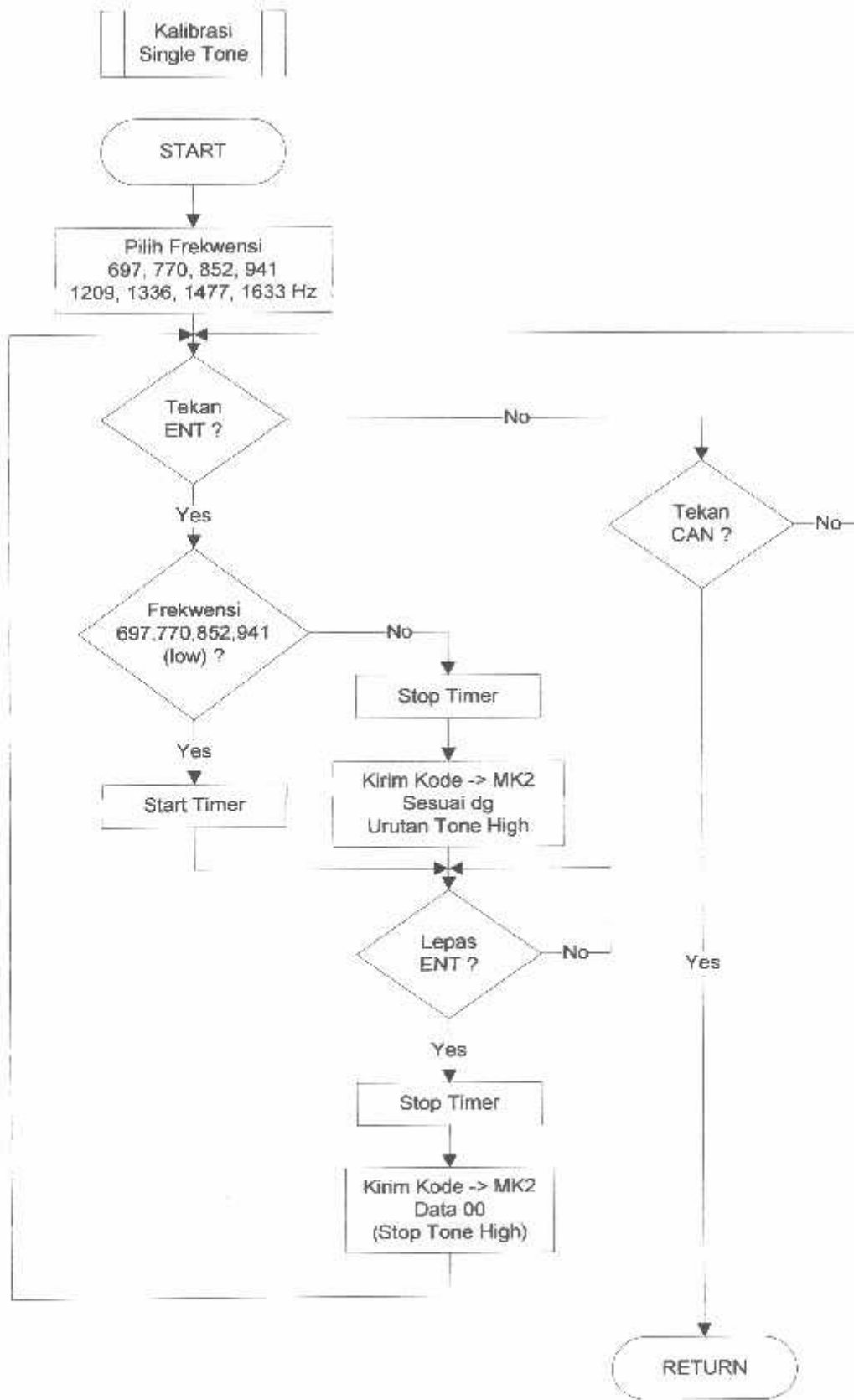


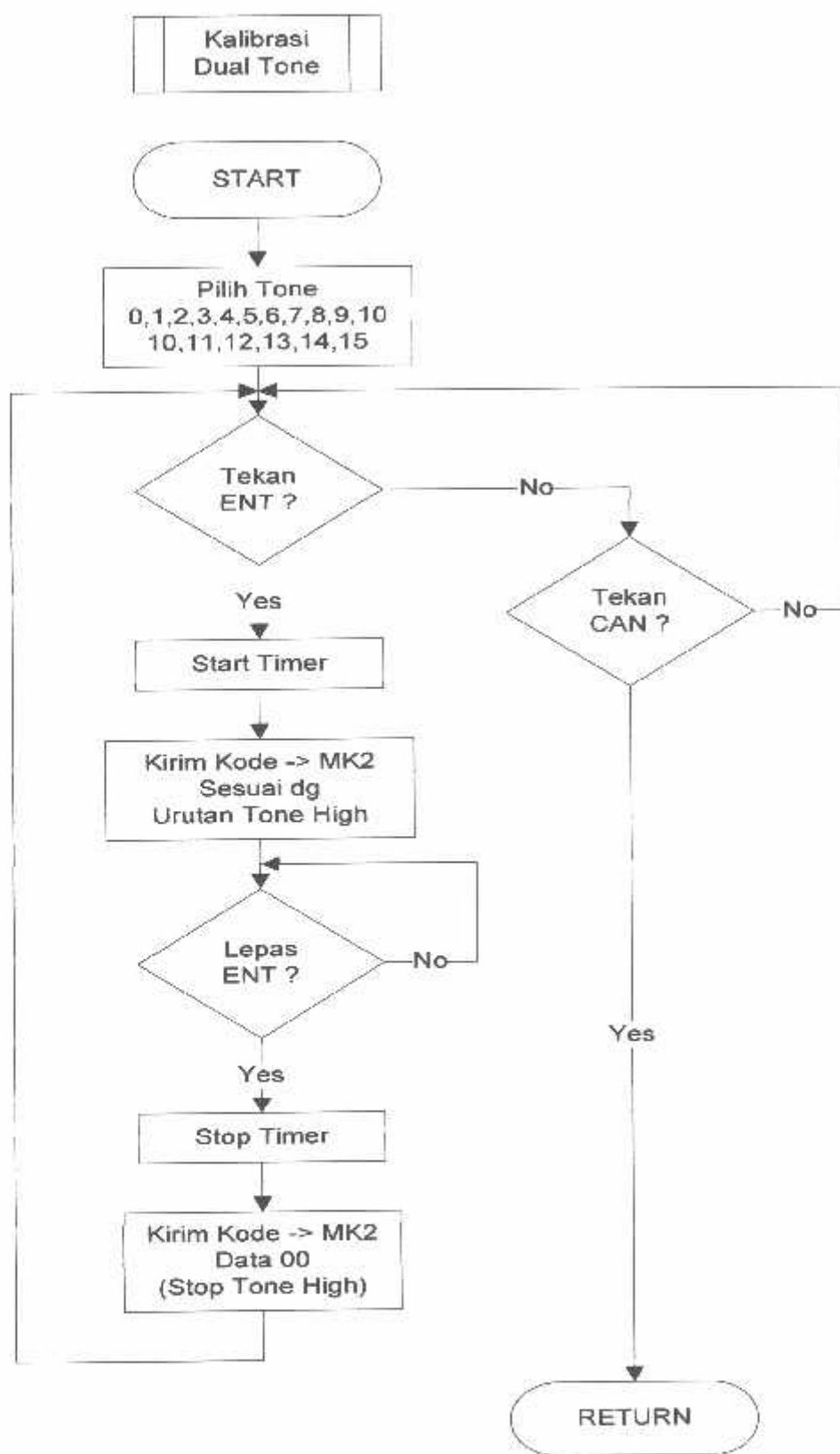
Isi Data Nama & No Telp



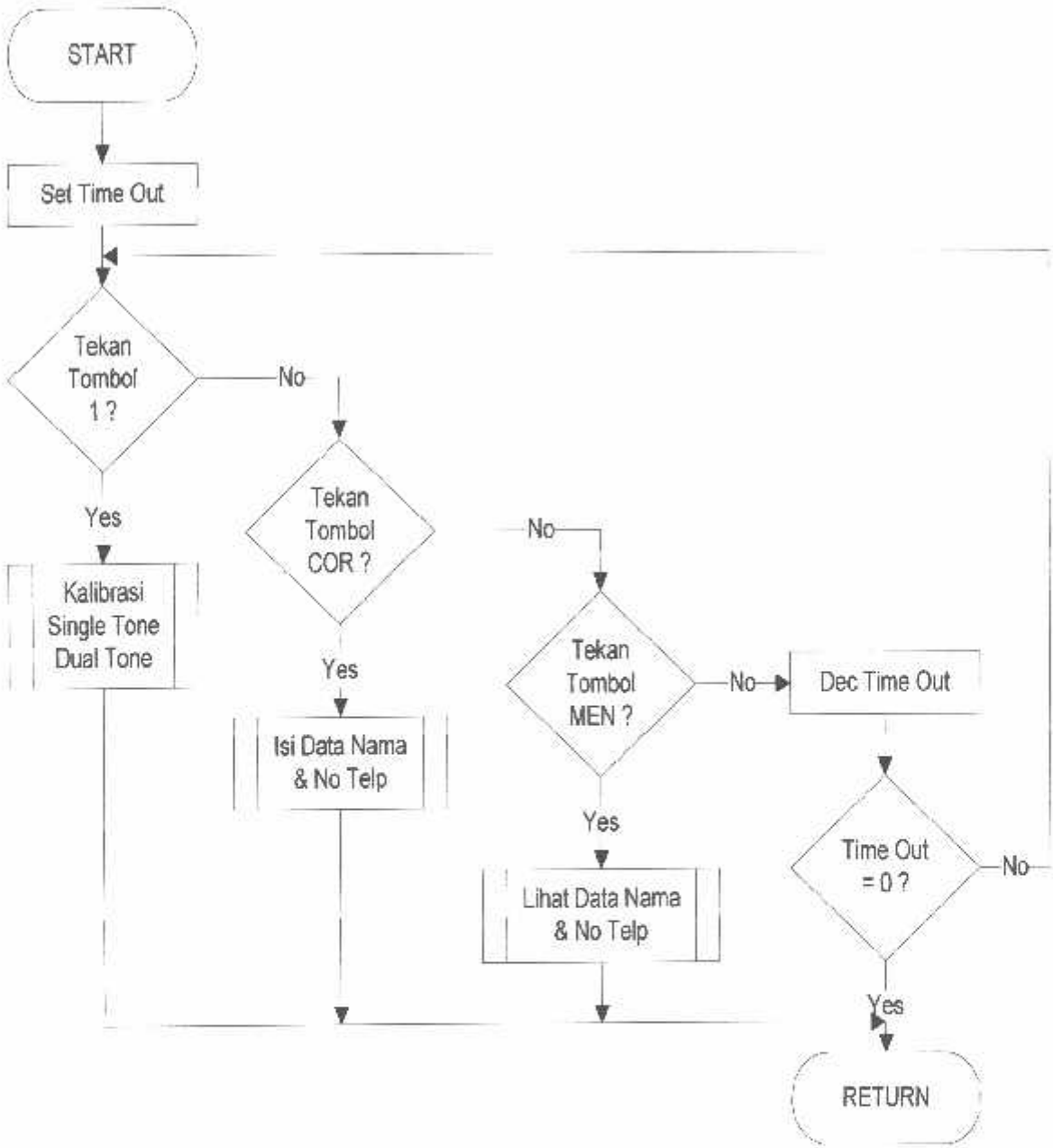
Lihat Data



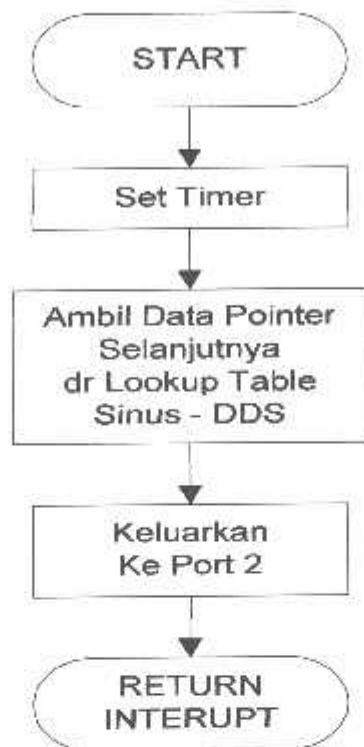


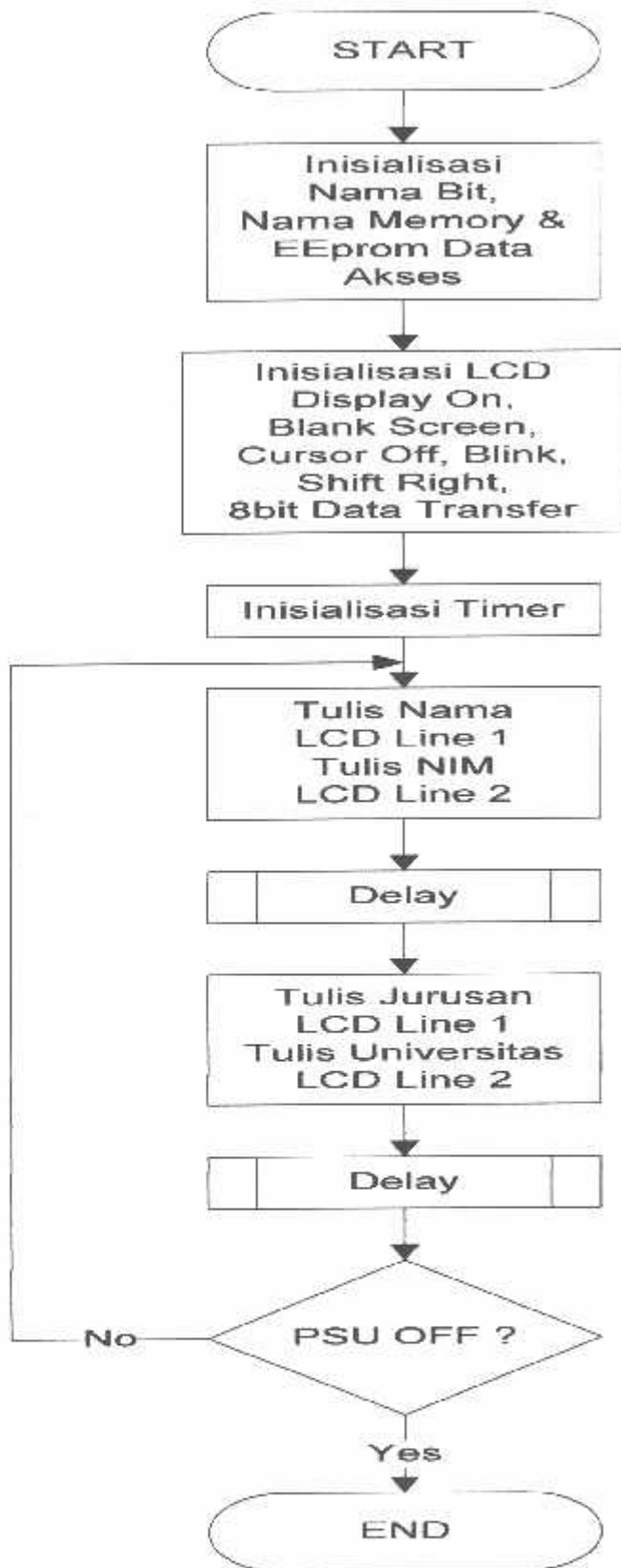


Delay



INTERUPT TIMER





frekwencje: 697, 770, 852, 941 Hz

```
org    00h
ljmp   init
```

```
org    0Bh
ljmp   drsine
```

```
Rest    Bit P3.0
Enb1     Bit P3.1
Slc0     Bit P3.4
Slc1     Bit P3.5
Slc2     Bit P3.6
Slc3     Bit P3.7
Stch     Bit 20h.0
Eecn     Data 96h           ; data eeprom control
Eecn     Equ 00001000b      ; bit eeprom enable (read)
Eewr     Equ 00010000b      ; bit eeprom write
Wtdg     Equ 00000010b      ; bit watchdog
Ch00     Equ 30h
Ch01     Equ 31h
Ch02     Equ 32h
Ch03     Equ 33h
Ch04     Equ 34h
Ch05     Equ 35h
Ch06     Equ 36h
Ch07     Equ 37h
Ch08     Equ 38h
Ch09     Equ 39h
Ch0A     Equ 3Ah
Ch0B     Equ 3Bh
Ch0C     Equ 3Ch
Ch0D     Equ 3Dh
Ch0E     Equ 3Eh
Ch0F     Equ 3Fh

Ch10     Equ 40h
Ch11     Equ 41h
Ch12     Equ 42h
Ch13     Equ 43h
Ch14     Equ 44h
Ch15     Equ 45h
Ch16     Equ 46h
Ch17     Equ 47h
Ch18     Equ 48h
Ch19     Equ 49h
Ch1A     Equ 4Ah
Ch1B     Equ 4Bh
Ch1C     Equ 4Ch
Ch1D     Equ 4Dh
Ch1E     Equ 4Eh
Ch1F     Equ 4Fh

Dtkp     Equ 50h
DtaL     Equ 51h
DtaH     Equ 52h
Char     Equ 53h
Cnt0     Equ 54h
Cnt1     Equ 55h
Cnt2     Equ 56h
Cnt3     Equ 57h
Tmc0     Equ 58h
Tmc1     Equ 59h
Colm     Equ 5Ah
Jum1     Equ 5Bh
Urot     Equ 5Ch
Dly0     Equ 5Dh
Dly1     Equ 5Eh
Dly2     Equ 5Fh
Dly3     Equ 60h
Dly4     Equ 61h
```

```

init:    lcall    lcd_in
         lcall    tmr_in
         mov     R1, #10

;
mulai:   mov     DPTR, #nama
         lcall    line1
         mov     Char, #16
         lcall    tulis
         mov     DPTR, #nim
         lcall    line2
         mov     Char, #16
         lcall    tulis
         lcall    delay3
         mov     DPTR, #jurs
         lcall    line1
         mov     Char, #16
         lcall    tulis
         mov     DPTR, #univ
         lcall    line2
         mov     Char, #16
         lcall    tulis
         lcall    delay3
         ljmp    mulai

;
isdt0:   lcall    lcdclr
         mov     DPTR, #tpisdt
         lcall    line1
         mov     Char, #16
         lcall    tulis
         lcall    tg_ips
         lcall    delay2

;
         lcall    lcdclr
         lcall    rstchr
         mov     Jm1, #16
         mov     Colm, #080h
isdt00:  mov     Dtkp, #10
         lcall    tg_tkn
         cjne    R0, #11, isdt01
         mov     SP, #07h
         ljmp    mulai
isdt01:  cjne    R0, #12, isdt02
         ljmp    isdt13           ; jump next section
isdt02:  lcall    senchr
isdt03:  mov     P0, Colm
         lcall    w_ins
         lcall    wr_chr
         lcall    bfcchr0
         mov     P0, Colm
         lcall    w_ins
         lcall    tg_ips
         mov     Tmo0, #10
         mov     Tmo1, 40
isdt04:  lcall    senkpd
         cjne    R0, #10, isdt05
         djnz    Tmo1, isdt04
         djnz    Tmo0, isdt04
         ljmp    isdt12
isdt05:  cjne    R0, #11, isdt06
         mov     SP, #07h
         ljmp    mulai
isdt06:  cjne    R0, #12, isdt07
         ljmp    isdt13           ; jump next section
isdt07:  cjne    R0, #13, isdt08
         ljmp    isdt04           ; not recommended
isdt08:  cjne    R0, #14, isdt09
         ljmp    isdt04           ; not recommended
isdt09:  cjne    R0, #15, isdt10
         ljmp    isdt04           ; not recommended
isdt10:  cjne    R0, #16, isdt11
         ljmp    isdt04           ; not recommended
isdt11:  lcall    senchr

```

```

jnb     Stc, isdt03
inc     Colm
djnz    Jum1, isdt05
ljmp     isdt13
isdt12: inc     Colm
        mov     P0, Colm
        lcall   w_ins
        djnz    Jum1, lompat
        ljmp     isdt13
lompat: ljmp     isdt00
;
isdt13: lcall   tg_ips
        mov     Jum1, #16
        mov     DPTR, #angka
        lcall   line2
isdt14: lcall   senkpd
        cjne    R0, #10, isdt15
        ljmp     isdt14
isdt15: cjne    R0, #11, isdt16
        mov     SP, #07h
        ljmp     mulai
isdt16: cjne    R0, #12, isdt17
        ljmp     isdt22
isdt17: cjne    R0, #13, isdt18
        ljmp     isdt14
isdt18: cjne    R0, #14, isdt19
        ljmp     isdt14
isdt19: cjne    R0, #15, isdt20
        ljmp     isdt14
isdt20: cjne    R0, #16, isdt21
        ljmp     isdt14
isdt21: mov     A, R0
        lcall   wr_chr
        lcall   bchr1
        lcall   tg_ips
        djnz    Jum1, isdt14
;
isdt22: lcall   cr_ksq
        lcall   wr_mem
        mov     SP, #07h
        ljmp     mulai
;
1hdt00: lcall   ledclr
        mov     DPTR, #tphiht
        lcall   line1
        mov     Char, #16
        lcall   tulis
        lcall   tg_ips
        lcall   delay2
        lcall   ledclr
        mov     Urut, #0
1hdt00: mov     DPTR, #0000h
        mov     A, Urut
        jz      1hdt03
        mov     Cnt1, Urut
1hdt01: mov     Cnt2, #32
1hdt02: inc     DPTR
        djnz    Cnt2, 1hdt02
        djnz    Cnt1, 1hdt01
1hdt03: lcall   rd_mem
        lcall   line1
        mov     P0, Ch00
        lcall   w_chr
        mov     P0, Ch01
        lcall   w_chr
        mov     P0, Ch02
        lcall   w_chr
        mov     P0, Ch03
        lcall   w_chr
        mov     P0, Ch04
        lcall   w_chr
        mov     P0, Ch05

```

; next section

```

lcall    w_ch
mov      P0,ChC6
lcall    w_ch
mov      P0,ChC7
lcall    w_ch
mov      P0,ChC8
lcall    w_ch
mov      P0,ChC9
lcall    w_ch
mov      P0,ChCA
lcall    w_ch
mov      P0,ChCB
lcall    w_ch
mov      P0,ChCC
lcall    w_ch
mov      P0,ChCD
lcall    w_ch
mov      P0,ChCE
lcall    w_ch
mov      P0,ChCF
lcall    w_ch

```

```

;
mov      DPTR,#angka
lcall    line2
mov      A,Ch10
lcall    wr_ch
mov      A,Ch11
lcall    wr_ch
mov      A,Ch12
lcall    wr_ch
mov      A,Ch13
lcall    wr_ch
mov      A,Ch14
lcall    wr_ch
mov      A,Ch15
lcall    wr_ch
mov      A,Ch16
lcall    wr_ch
mov      A,Ch17
lcall    wr_ch
mov      A,Ch18
lcall    wr_ch
mov      A,Ch19
lcall    wr_ch
mov      A,Ch1A
lcall    wr_ch
mov      A,Ch1B
lcall    wr_ch
mov      A,Ch1C
lcall    wr_ch
mov      A,Ch1D
lcall    wr_ch
mov      A,Ch1E
lcall    wr_ch
mov      A,Ch1F
lcall    wr_ch

```

```

;
lcall    tg_ips
1hdt04: lcall    senkpd
        cjne    R0,#11,1hdt05
        mov     SP,#07h
        ljmp    mulai
1hdt05: cjne    R0,#12,1hdt06
        ljmp    dialno
1hdt06: cjne    R0,#15,1hdt08
        dec     Urut
        mov     A,Urut
        cjne    A,#255,1hdt07
        mov     Urut,#0
1hdt07: ljmp    1hdt00
1hdt08: cjne    R0,#16,1hdt10
        inc     Urut

```

```

mov     A,Urut
cjne    A,#61,lhdt09
mov     Urut,#60
lhdt09: ljmp     lhdt00
lhdt10: cjne    R0,#0,lhdt04
        lcall   t1sdel
        mov     DPTR,#0000h
        mov     A,Urut
        jz      lhdt13
        mov     Cnt1,Urut
lhdt11:  mov     Cnt2,#32
lhdt12:  inc     DPTR
        djnz    Cnt2,lhdt12
        djnz    Cnt1,lhdt11
lhdt13:  lcall   rstchr
        lcall   wr_mem
        ljmp    lhdt00
;
kalbrs:  lcall   lodelr
klbr00:  mov     DPTR,#sigltn
        lcall   line1
        mov     Char,#16
        lcall   tulis
        lcall   tg_lps
klbr01:  lcall   senkpd
        cjne    R0,#11,klbr02
        mov     SP,#07h
        ljmp    mulai
klbr02:  cjne    R0,#12,klbr03
        ljmp    klbr07
klbr03:  cjne    R0,#16,klbr01
        mov     DPTR,#dualtn
        lcall   line1
        mov     Char,#16
        lcall   tulis
        lcall   tg_lps
klbr04:  lcall   senkpd
        cjne    R0,#11,klbr05
        mov     SP,#07h
        ljmp    mulai
klbr05:  cjne    R0,#12,klbr06
        ljmp    klbr26
klbr06:  cjne    R0,#15,klbr04
        ljmp    klbr00
;
klbr07:  mov     R5,#0
klbr08:  cjne    R5,#0,klbr09
        mov     DPTR,#sgltn0
klbr09:  cjne    R5,#1,klbr10
        mov     DPTR,#sgltn1
klbr10:  cjne    R5,#2,klbr11
        mov     DPTR,#sgltn2
klbr11:  cjne    R5,#3,klbr12
        mov     DPTR,#sgltn3
klbr12:  cjne    R5,#4,klbr13
        mov     DPTR,#sgltn4
klbr13:  cjne    R5,#5,klbr14
        mov     DPTR,#sgltn5
klbr14:  cjne    R5,#6,klbr15
        mov     DPTR,#sgltn6
klbr15:  cjne    R5,#7,klbr16
        mov     DPTR,#sgltn7
klbr16:  lcall   line2
        mov     Char,#16
        lcall   tulis
        lcall   tg_lps
klbr17:  lcall   senkpd
        cjne    R0,#11,klbr18
        ljmp    kalbrs
klbr18:  cjne    R0,#12,klbr19
        ljmp    klbr20
klbr19:  ljmp    klbr21

```

```

klbr20: cjne R5,#0,sentn0
        mov  R2,#0
        mov  R3,#4
        lcall cek00
        lcall tq_lps
        setb TR0
        ljmp sentn7
sentn0: cjne R5,#1,sentn1
        mov  R2,#1
        mov  R3,#4
        lcall cek00
        lcall tq_lps
        setb TR0
        ljmp sentn7
sentn1: cjne R5,#2,sentn2
        mov  R2,#2
        mov  R3,#4
        lcall cek00
        lcall tq_lps
        setb TR0
        ljmp sentn7
sentn2: cjne R5,#3,sentn3
        mov  R2,#3
        mov  R3,#4
        lcall tq_lps
        lcall cek00
        setb TR0
        ljmp sentn7
sentn3: cjne R5,#4,sentn4
        mov  R2,#0
        mov  R3,#0
        lcall cek00
        ljmp sentn7
sentn4: cjne R5,#5,sentn5
        mov  R2,#0
        mov  R3,#1
        lcall cek00
        ljmp sentn7
sentn5: cjne R5,#6,sentn6
        mov  R2,#0
        mov  R3,#2
        lcall cek00
        ljmp sentn7
sentn6: cjne R5,#7,sentn7
        mov  R2,#0
        mov  R3,#3
        lcall cek00
sentn7: lcall senkpd
        lcall delay0
        cjne R0,#11,sentn7
        clr  TR0
        lcall tnhist
        lcall tq_lps
        ljmp klbr17
;
klbr21: cjne R0,#15,klbr23
        dec  R5
        cjne R5,#255,klbr22
        mov  R5,#0
klbr22: ljmp klbr08
klbr23: cjne R0,#16,klbr25
        inc  R5
        cjne R5,#9,klbr24
        mov  R5,#7
klbr24: ljmp klbr08
klbr25: ljmp klbr17
;
klbr26: mov  DPTR,#tptone
        mov  Char,#16
        lcall line2
        lcall tuiis
        mov  R5,#00

```

```

klbr27: cjne R5, #0, klbr28
        mov  DPTR, #duatn0
        mov  R6, #01
klbr28: cjne R5, #01, klbr29
        mov  DPTR, #duatn1
        mov  R6, #02
klbr29: cjne R5, #02, klbr30
        mov  DPTR, #duatn2
        mov  R6, #03
klbr30: cjne R5, #03, klbr31
        mov  DPTR, #duatn3
        mov  R6, #04
klbr31: cjne R5, #04, klbr32
        mov  DPTR, #duatn4
        mov  R6, #05
klbr32: cjne R5, #05, klbr33
        mov  DPTR, #duatn5
        mov  R6, #06
klbr33: cjne R5, #06, klbr34
        mov  DPTR, #duatn6
        mov  R6, #07
klbr34: cjne R5, #07, klbr35
        mov  DPTR, #duatn7
        mov  R6, #08
klbr35: cjne R5, #08, klbr36
        mov  DPTR, #duatn8
        mov  R6, #09
klbr36: cjne R5, #09, klbr37
        mov  DPTR, #duatn9
        mov  R6, #10
klbr37: cjne R5, #10, klbr38
        mov  DPTR, #duatnA
        mov  R6, #11
klbr38: cjne R5, #11, klbr39
        mov  DPTR, #duatnB
        mov  R6, #12
klbr39: cjne R5, #12, klbr40
        mov  DPTR, #duatnC
        mov  R6, #13
klbr40: cjne R5, #13, klbr41
        mov  DPTR, #duatnD
        mov  R6, #14
klbr41: cjne R5, #14, klbr42
        mov  DPTR, #duatnE
        mov  R6, #15
klbr42: cjne R5, #15, klbr43
        mov  DPTR, #duatnF
        mov  R6, #00
klbr43: lcall line1
        mov  Char, #16
        lcall tulis
        mov  DPTR, #angka
        mov  P0, #0CAh
        lcall w_ins
        mov  A, R6
        mov  B, #10
        div  A3
        lcall wr_chr
        mov  A, B
        lcall wr_chr
        mov  P0, #0D0h
        lcall w_ins
        lcall tg_lps
klbr44: lcall scnkpdt
        cjne R0, #11, klbr45
        ljmp kalbrs
klbr45: cjne R0, #12, klbr46
        lcall tg_lps
        ljmp klbr50
klbr46: cjne R0, #15, klbr48
        dec  R5
        cjne R5, #255, klbr47

```

```

mov R5,#0
klbr47: ljmp klbr27
klbr48: cjne R0,#16,klbr44
inc R5
cjne R5,#16,klbr49
mov R5,#15
klbr49: ljmp klbr27
;
klbr50: mov DPTR,#lokup1
mov A,R5
movc A,@A+DPTR
mov B,#10
div AB
mov R2,A
mov R3,B
lcall cek00
setb TR0
klbr51: lcall senkpd
lcall delay0
cjne R0,#11,klbr51
clr TR0
lcall tmhist
ljmp klbr27
;
dialno: mov A,Ch10
lcall sctone
mov A,Ch11
lcall sctone
mov A,Ch12
lcall sctone
mov A,Ch13
lcall sctone
mov A,Ch14
lcall sctone
mov A,Ch15
lcall sctone
mov A,Ch16
lcall sctone
mov A,Ch17
lcall sctone
mov A,Ch18
lcall sctone
mov A,Ch19
lcall sctone
mov A,Ch1A
lcall sctone
mov A,Ch1B
lcall sctone
mov A,Ch1C
lcall sctone
mov A,Ch1D
lcall sctone
mov A,Ch1E
lcall sctone
mov A,Ch1F
lcall sctone
ljmp lhdt00
;
cr_ksg: lcall strdmm
mov DPTR,#0000h
crksg0: movx A,@DPTR
cjne A,#' ',crksg1
ljmp crksg3
crksg1: mov Cnt3,#32
crksg2: inc DPTR
djnz Cnt3,crksg2
ljmp crksg0
crksg3: lcall enrdmm
ret
;
bfchr0: mov R4,Calm
cjne R4,#08Ch,bfch00

```

```

mov Ch0,A
bfch00: cjne R4,#081h,bfch01
mov Ch01,A
bfch01: cjne R4,#082h,bfch02
mov Ch02,A
bfch02: cjne R4,#083h,bfch03
mov Ch03,A
bfch03: cjne R4,#084h,bfch04
mov Ch04,A
bfch04: cjne R4,#085h,bfch05
mov Ch05,A
bfch05: cjne R4,#086h,bfch06
mov Ch06,A
bfch06: cjne R4,#087h,bfch07
mov Ch07,A
bfch07: cjne R4,#088h,bfch08
mov Ch08,A
bfch08: cjne R4,#089h,bfch09
mov Ch09,A
bfch09: cjne R4,#08Ah,bfch0A
mov Ch0A,A
bfch0A: cjne R4,#08Bh,bfch0B
mov Ch0B,A
bfch0B: cjne R4,#08Ch,bfch0C
mov Ch0C,A
bfch0C: cjne R4,#08Dh,bfch0D
mov Ch0D,A
bfch0D: cjne R4,#08Eh,bfch0E
mov Ch0E,A
bfch0E: cjne R4,#08Fh,bfch0F
mov Ch0F,A
bfch0F: ret
;
bfchl1: mov R4,Jum1
cjne R4,#16,bfchl10
mov Ch10,R0
bfchl10: cjne R4,#15,bfchl11
mov Ch11,R0
bfchl11: cjne R4,#14,bfchl12
mov Ch12,R0
bfchl12: cjne R4,#13,bfchl13
mov Ch13,R0
bfchl13: cjne R4,#12,bfchl14
mov Ch14,R0
bfchl14: cjne R4,#11,bfchl15
mov Ch15,R0
bfchl15: cjne R4,#10,bfchl16
mov Ch16,R0
bfchl16: cjne R4,#09,bfchl17
mov Ch17,R0
bfchl17: cjne R4,#08,bfchl18
mov Ch18,R0
bfchl18: cjne R4,#07,bfchl19
mov Ch19,R0
bfchl19: cjne R4,#06,bfchl1A
mov Ch1A,R0
bfchl1A: cjne R4,#05,bfchl1B
mov Ch1B,R0
bfchl1B: cjne R4,#04,bfchl1C
mov Ch1C,R0
bfchl1C: cjne R4,#03,bfchl1D
mov Ch1D,R0
bfchl1D: cjne R4,#02,bfchl1E
mov Ch1E,R0
bfchl1E: cjne R4,#01,bfchl1F
mov Ch1F,R0
bfchl1F: ret
;
stone: mov DPTR,#lookupC
movc A,@A+DPTR
cjne A,#99,stone0
ljmp stone1

```

```

stone0: mov     B, #10
        div     A3
        mov     R2, A
        mov     R3, B
        lcall   cek00
        setb    TR0
        lcall   dltone
        mov     Dly1, #25
        lcall   dltone
        clr     TR0
        lcall   lnhist
        lcall   dltone
stone1: ret
;
cek00:  cjne    R2, #0, cek01
        mov     DtaL, #0E9h          ; 65536-(1jt/697)
        mov     DtaH, #0FEh          ; 65536-1434=64102
cek01:  cjne    R2, #1, cek02
        mov     DtaL, #004h          ; 65536-(1jt/770)
        mov     DtaH, #0FFh          ; 65536-1299=64237
cek02:  cjne    R2, #2, cek03
        mov     DtaL, #01Dh          ; 65536-(1jt/852)
        mov     DtaH, #0FFh          ; 65536-1174=64362
cek03:  cjne    R2, #3, cek04
        mov     DtaL, #033h          ; 65536-(1jt/941)
        mov     DtaH, #0FFh          ; 65536-1063=64473
cek04:  cjne    R3, #0, cek05
        setb    Slc0                ; tone 1633
        setb    Slc1                ; tone 1477
        setb    Slc2                ; tone 1336
        clr     Slc3                ; tone 1209
cek05:  cjne    R3, #1, cek06
        setb    Slc0                ; tone 1633
        setb    Slc1                ; tone 1477
        clr     Slc2                ; tone 1336
        setb    Slc3                ; tone 1209
cek06:  cjne    R3, #2, cek07
        setb    Slc0                ; tone 1633
        clr     Slc1                ; tone 1477
        setb    Slc2                ; tone 1336
        setb    Slc3                ; tone 1209
cek07:  cjne    R3, #3, cek08
        clr     Slc0                ; tone 1633
        setb    Slc1                ; tone 1477
        setb    Slc2                ; tone 1336
        setb    Slc3                ; tone 1209
cek08:  ret
;
rstchr: mov     Ch00, #' '
        mov     Ch01, #' '
        mov     Ch02, #' '
        mov     Ch03, #' '
        mov     Ch04, #' '
        mov     Ch05, #' '
        mov     Ch06, #' '
        mov     Ch07, #' '
        mov     Ch08, #' '
        mov     Ch09, #' '
        mov     Ch0A, #' '
        mov     Ch0B, #' '
        mov     Ch0C, #' '
        mov     Ch0D, #' '
        mov     Ch0E, #' '
        mov     Ch0F, #' '

        mov     Ch10, #10
        mov     Ch11, #10
        mov     Ch12, #10
        mov     Ch13, #10
        mov     Ch14, #10
        mov     Ch15, #10
        mov     Ch16, #10

```

```

mov     Ch18,#10
mov     Ch19,#10
mov     Ch1A,#10
mov     Ch1B,#10
mov     Ch1C,#10
mov     Ch1D,#10
mov     Ch1E,#10
mov     Ch1F,#10

;
line1:  mov     P0,#080h
        lcall   w_ins
        ret

;
line2:  mov     P0,#0C0h
        lcall   w_ins
        ret

;
tulis:  clr     A
        lcall   wr_chr
        inc     DPTR
        djnz    Char,tulis
        ret

;
wr_chr:  movc    A,$A,DPTR
        mov     P0,A
        lcall   w_chr
        ret

;
w_ins:  clr     Enbl
        clr     Rest
        setb    Enbl
        clr     Enbl
        lcall   delay0
        lcall   delay0
        ret

;
w_chr:  clr     Enbl
        setb    Rest
        setb    Enbl
        clr     Enbl
        lcall   delay0
        lcall   delay0
        ret

;
led_in: lcall   delay1
        mov     P0,#01h           ; Display Clear
        lcall   w_ins
        mov     P0,#38h           ; Function Set
        lcall   w_ins
        mov     P0,#0Eh           ; Display On, Cursor, Blink
        lcall   w_ins
        mov     P0,#06h           ; Entry Mode
        lcall   w_ins
        mov     P0,#02h           ; Cursor Home
        lcall   w_ins
        ret

;
ledclr: mov     P0,#01h           ; Display Clear
        lcall   w_ins
        lcall   delay0
        lcall   delay0
        lcall   delay0
        lcall   delay0
        lcall   delay0
        ret

;
tmr_in: lcall   delay1
        lcall   delay1
        mov     TMOD,#10h
        mov     TLO,#000h
        mov     TH0,#000h

```

```

        setb    ET0
        clr     TF0
        setb    EA
        mov     P2, #00h
        ret

;
drsine: mov     TL0, 0taL
        mov     TH0, 0taH
        clr     TF0
        mov     DPTR, #lookup2
        mov     A, R1
        movc    A, @A+DPTR
        mov     P2, A
        djnz    R1, dsine
        mov     R1, #10
dsine:  reti

;
tnhist: setb    S1c0           ; tone high stop
        setb    S1c1
        setb    S1c2
        setb    S1c3
        ret

;
stwrmm: orl     Eecn, #Eeen
        orl     Eecn, #Eewr
        ret

;
enwrmm: xrl     Eecn, #Eewr
        xrl     Eecn, #Eeen
        ret

;
strdmm: orl     Eecn, #Eeen
        ret

;
enrdmm: xrl     Eecn, #Eeen
        ret

;
wl_wr:  mov     A, Eecn
        anl     A, #Wtdg
        jz      wt_wr
        ret

;
tisdel: lcall   lddclr
        mov     DPTR, #diting
        lcall   line2
        mov     Char, #16
        lcall   tulis
        ret

;
wr_mem: lcall   stwrmm
        mov     A, Ch00
        movx    @DPTR, A
        lcall   wt_wr
        inc     DPTR
        mov     A, Ch01
        movx    @DPTR, A
        lcall   wt_wr
        inc     DPTR
        mov     A, Ch02
        movx    @DPTR, A
        lcall   wt_wr
        inc     DPTR
        mov     A, Ch03
        movx    @DPTR, A
        lcall   wt_wr
        inc     DPTR
        mov     A, Ch04
        movx    @DPTR, A
        lcall   wt_wr
        inc     DPTR
        mov     A, Ch05
        movx    @DPTR, A

```

```

lcall wt_wr
inc DPTR
mov A, Ch06
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch07
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch08
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch09
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch0A
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch0B
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch0C
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch0D
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch0E
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch0F
movx @DPTR, A
lcall wt_wr

inc DPTR
mov A, Ch10
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch11
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch12
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch13
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch14
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch15
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch16
movx @DPTR, A
lcall wt_wr
inc DPTR
mov A, Ch17

```

```

movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch18
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch19
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch1A
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch1B
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch1C
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch1D
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch1E
movx    @DPTR,A
lcall   wt_wr
inc     DPTR
mov     A,Ch1F
movx    @DPTR,A
lcall   wt_wr
lcall   enwxmm
ret

```

```

;
rd_mem: lcall   strcmm
movx    A,@DPTR
mov     Ch00,A
inc     DPTR
movx    A,@DPTR
mov     Ch01,A
inc     DPTR
movx    A,@DPTR
mov     Ch02,A
inc     DPTR
movx    A,@DPTR
mov     Ch03,A
inc     DPTR
movx    A,@DPTR
mov     Ch04,A
inc     DPTR
movx    A,@DPTR
mov     Ch05,A
inc     DPTR
movx    A,@DPTR
mov     Ch06,A
inc     DPTR
movx    A,@DPTR
mov     Ch07,A
inc     DPTR
movx    A,@DPTR
mov     Ch08,A
inc     DPTR
movx    A,@DPTR
mov     Ch09,A
inc     DPTR
movx    A,@DPTR
mov     Ch0A,A
inc     DPTR
movx    A,@DPTR

```

```

mov     Ch08,A
inc     DPTR
movx    A,@DPTR
mov     Ch0C,A
inc     DPTR
movx    A,@DPTR
mov     Ch0D,A
inc     DPTR
movx    A,@DPTR
mov     Ch0E,A
inc     DPTR
movx    A,@DPTR
mov     Ch0F,A

```

```

inc     DPTR
movx    A,@DPTR
mov     Ch10,A
inc     DPTR
movx    A,@DPTR
mov     Ch11,A
inc     DPTR
movx    A,@DPTR
mov     Ch12,A
inc     DPTR
movx    A,@DPTR
mov     Ch13,A
inc     DPTR
movx    A,@DPTR
mov     Ch14,A
inc     DPTR
movx    A,@DPTR
mov     Ch15,A
inc     DPTR
movx    A,@DPTR
mov     Ch16,A
inc     DPTR
movx    A,@DPTR
mov     Ch17,A
inc     DPTR
movx    A,@DPTR
mov     Ch18,A
inc     DPTR
movx    A,@DPTR
mov     Ch19,A
inc     DPTR
movx    A,@DPTR
mov     Ch1A,A
inc     DPTR
movx    A,@DPTR
mov     Ch1B,A
inc     DPTR
movx    A,@DPTR
mov     Ch1C,A
inc     DPTR
movx    A,@DPTR
mov     Ch1D,A
inc     DPTR
movx    A,@DPTR
mov     Ch1E,A
inc     DPTR
movx    A,@DPTR
mov     Ch1F,A
lcall   enrdmm
ret

```

```

;
senkpd: mov     R0,#10
        lcall   delay0
        lcall   delay0
coll:   mov     P1,#111111110b
        mov     A,P1
c1b1:   cjne    A,#111101110b,c1b2
        mov     R0,#1

```

```

c1b2:  cjne  A,#11011110b,c1b3
        mov  R0,#2
c1b3:  cjne  A,#10111110b,c1b4
        mov  R0,#3
c1b4:  cjne  A,#01111110b,col2
        mov  R0,#13
;
col2:  mov  P1,#11111101b
        mov  A,P1
c2b1:  cjne  A,#11101101b,c2b2
        mov  R0,#4
c2b2:  cjne  A,#11011101b,c2b3
        mov  R0,#5
c2b3:  cjne  A,#10111101b,c2b4
        mov  R0,#6
c2b4:  cjne  A,#01111101b,col3
        mov  R0,#14
;
col3:  mov  P1,#11110111b
        mov  A,P1
c3b1:  cjne  A,#11101011b,c3b2
        mov  R0,#7
c3b2:  cjne  A,#11011011b,c3b3
        mov  R0,#8
c3b3:  cjne  A,#10111011b,c3b4
        mov  R0,#9
c3b4:  cjne  A,#01111011b,col4
        mov  R0,#15
;
col4:  mov  P1,#11110111b
        mov  A,P1
c4b1:  cjne  A,#11100111b,c4b2
        mov  R0,#11
c4b2:  cjne  A,#11010111b,c4b3
        mov  R0,#0
c4b3:  cjne  A,#10110111b,c4b4
        mov  R0,#12
c4b4:  cjne  A,#01110111b,back
        mov  R0,#16
back:  ret
;
tg_tkn: lcall  senkpd
tgtkn0: cjne  R0,#16,tgtkn1
        ljmp  tg_tkn
tgtkn1: cjne  R0,#15,tgtkn2
        ljmp  tg_tkn
tgtkn2: cjne  R0,#14,tgtkn3
        ljmp  tg_tkn
tgtkn3: cjne  R0,#13,tgtkn4
        ljmp  tg_tkn
tgtkn4: cjne  R0,#10,tgtkn5
        ljmp  tg_tkn
tgtkn5: ret
;
tg_lps: lcall  senkpd
        cjne  R0,#10,tg_lps
        ret
;
senchr: clr  Stch
        mov  A,R0
        mov  B,Dtkp
        clr  C
        subb A,B
        jnz  senchr0
        ljmp senchr1
senchr0: setb Stch
        ljmp senchr3
senchr1: djnz R1,senchr2
        mov  R1,Cnt0
        ljmp senchr3
senchr2: inc  DPTR
        ljmp senchr4

```

```

scchr3: mov     Dtkp, #0
        mov     A, Dtkp
        lcall   skimpk
scchr4: clr     A
        ret

;
skimpk: cjne    A, #1, skimp0
        mov     DPTR, #skimpk0
        mov     Cnt0, #2
        mov     R1, Cnt0
skimp0: cjne    A, #2, skimp1
        mov     DPTR, #skimpk1
        mov     Cnt0, #4
        mov     R1, Cnt0
skimp1: cjne    A, #3, skimp2
        mov     DPTR, #skimpk2
        mov     Cnt0, #4
        mov     R1, Cnt0
skimp2: cjne    A, #4, skimp3
        mov     DPTR, #skimpk3
        mov     Cnt0, #4
        mov     R1, Cnt0
skimp3: cjne    A, #5, skimp4
        mov     DPTR, #skimpk4
        mov     Cnt0, #4
        mov     R1, Cnt0
skimp4: cjne    A, #6, skimp5
        mov     DPTR, #skimpk5
        mov     Cnt0, #4
        mov     R1, Cnt0
skimp5: cjne    A, #7, skimp6
        mov     DPTR, #skimpk6
        mov     Cnt0, #5
        mov     R1, Cnt0
skimp6: cjne    A, #8, skimp7
        mov     DPTR, #skimpk7
        mov     Cnt0, #4
        mov     R1, Cnt0
skimp7: cjne    A, #9, skimp8
        mov     DPTR, #skimpk8
        mov     Cnt0, #5
        mov     R1, Cnt0
skimp8: cjne    A, #0, skimp9
        mov     DPTR, #skimpk9
        mov     Cnt0, #2
        mov     R1, Cnt0
skimp9: ret
;
delay0: djnz    Dly0, delay0
        ret
;
delay1: lcall   scnkp0
        cjne    R0, #01, dely10
        ljmp    kalbrs
dely10: cjne    R0, #13, dely11
        ljmp    lsidta
dely11: cjne    R0, #14, dely12
        ljmp    lhtdta
dely12: djnz    Dly1, delay1
        ret
;
delay2: mov     Dly2, #5
dely2:  lcall   delay1
        djnz    Dly2, dely2
        ret
;
delay3: mov     Dly3, #30
dely3:  lcall   delay1
        djnz    Dly3, dely3
        ret
;
altone: lcall   delay0

```

[illegible]

end

end

```

;   frekwensi: 1209, 1336, 1477, 1633 Hz
;
;   org      00h
;   ljmp     init
;
;   org      0Bh
;   ljmp     drsine
;
;   DtaL     Equ 30h
;   DtaH     Equ 31h
;   Dly0     Equ 40h
;   Dly1     Equ 41h
;   Dly2     Equ 42h
;
init:   acall   tmr_in
        mov     DPTR,#lookup
        mov     R0,#10
;
cek00:  mov     A,P1
        anl     A,#15
        cjne    A,#00001111b,cek01
        mov     P2,#00h
        clr     TR0
        sjmp    cek05
cek01:  setb    TR0
        cjne    A,#00001110b,cek02
        mov     DtaL,#062h      ; 65536-(1jt/1209)
        mov     DtaH,#0FFh      ; 65536-827=64709
cek02:  cjne    A,#00001101b,cek03
        mov     DtaL,#072h      ; 65536-(1jt/1336)
        mov     DtaH,#0FFh      ; 65536-749=64787
cek03:  cjne    A,#00001011b,cek04
        mov     DtaL,#080h      ; 65536-(1jt/1477)
        mov     DtaH,#0FFh      ; 65536-677=64859
cek04:  cjne    A,#00000111b,cek05
        mov     DtaL,#08Dh      ; 65536-(1jt/1633)
        mov     DtaH,#0FFh      ; 65536-612=64924
cek05:  ljmp    cek00
;
drsine: mov     TL0,DtaL
        mov     TH0,DtaH
        clr     TF0
        mov     A,R0
        movc    A,@A+DPTR
        mov     P2,A
        djnz    R0,drsine
        mov     R0,#10
dsine:  reti
;
tmr_in: acall    delay1
        acall    delay1
        mov     TMOD,#11h
        mov     TL0,#000h
        mov     TH0,#000h
        setb    ET0
        clr     TF0
        setb    EA
        ret
;
delay0: djnz    Dly0,delay0
        ret
;
delay1: acall    delay0
        djnz    Dly1,delay1
        ret
;
lookup: DB      000,003,025,125,225,247,250,247,225,125,025
;
        end

```

